

Житомирський державний університет імені Івана Франка
Кафедра прикладної математики та інформатики

В.В. Міхєєв

Лабораторні роботи з методики навчання інформатики

Методичний посібник

Житомир – 2006

ЗМІСТ

ПРОГРАМА КУРСУ "МЕТОДИКА НАВЧАННЯ ІНФОРМАТИКИ"	4
ІНСТРУКТИВНО-МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ЩОДО ВИВЧЕННЯ ІНФОРМАТИКИ У НОВОМУ 2006/2007 НАВЧАЛЬНОМУ РОЦІ	5
МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ВИВЧЕННЯ КУРСУ "МЕТОДИКА НАВЧАННЯ ІНФОРМАТИКИ"	14
ПРОГРАМА КУРСУ "ОСНОВ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ" ДЛЯ ЗАГАЛЬНООСВІТНЬОЇ ШКОЛИ	16
ТЕМАТИЧНЕ ПЛАНУВАННЯ НАВЧАЛЬНОГО МАТЕРІАЛУ ТА ЗМІСТ НАВЧАННЯ	17
ОРІЄНТОВНЕ ПОУРОЧНЕ ПЛАНУВАННЯ КУРСУ "ОСНОВИ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ" 1 варіант (для комп'ютерів, які працюють під управлінням ОС Windows)	22
ОРІЄНТОВНЕ ПОУРОЧНЕ ПЛАНУВАННЯ КУРСУ "ОСНОВИ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ" 2 варіант (для комп'ютерів, які працюють під управлінням операційної системи MS-DOS)	25
РЕКОМЕНДОВАНА ЛІТЕРАТУРА	27
ПОЛОЖЕННЯ ПРО РЕЙТИНГОВУ СИСТЕМУ ОЦІНКИ ЗНАНЬ, ВМІНЬ ТА НАВИЧОК СТУДЕНТІВ ПРИ ВИВЧЕННІ КУРСУ "МЕТОДИКА НАВЧАННЯ ІНФОРМАТИКИ"	29
ЛАБОРАТОРНІ РОБОТИ	30
ЛАБОРАТОРНА РОБОТА №1 (4 год.)	30
Календарно-тематичний план вчителя інформатики	30
ЛАБОРАТОРНА РОБОТА №2 (4 год.)	33
Конспект уроку вчителя інформатики	33
ЛАБОРАТОРНА РОБОТА №3 (4 год.)	38
Програмне забезпечення ПЕОМ при навчанні інформатики	38
ЛАБОРАТОРНА РОБОТА №4 (4 год.)	42
Методика вивчення арифметичних, логічних та фізичних основ ЕОМ	42
ЛАБОРАТОРНА РОБОТА №5 (4 год.)	46
Методика вивчення інформаційної системи	46
ЛАБОРАТОРНА РОБОТА №6 (2 год.)	52
Методика вивчення операційної системи MS DOS	52
ЛАБОРАТОРНА РОБОТА №7 (4 год.)	56
Методика вивчення операційної системи MS Windows	56
ЛАБОРАТОРНА РОБОТА №8 (4 год.)	60
Методика вивчення основ роботи з дисками	60
ЛАБОРАТОРНА РОБОТА №9 (4 год.)	63
Методика вивчення глобальної мережі Інтернет та її можливостей	63
ЛАБОРАТОРНА РОБОТА №10 (10 год.)	65
Методика вивчення текстового редактора	65
ЛАБОРАТОРНА РОБОТА №11 (2 год.)	84
Методика вивчення графічного редактора	84
ЛАБОРАТОРНА РОБОТА №12 (8 год.)	87
Методика вивчення електронних таблиць	87
ЛАБОРАТОРНА РОБОТА №13 (8 год.)	106
Методика вивчення баз даних	106
ЛАБОРАТОРНА РОБОТА №14 (2 год.)	129
Навчальна алгоритмічна мова.	129

ЛАБОРАТОРНА РОБОТА №15 (4 год.)	133
Мова програмування BASIC	133
Лінійне програмування числових виразів	133
ЛАБОРАТОРНА РОБОТА №16 (2 год.)	137
Лінійне програмування графічних об'єктів	137
ЛАБОРАТОРНА РОБОТА №17–18 (8 год.)	139
Розгалужене та циклічне програмування	139
ЛАБОРАТОРНА РОБОТА №19 (4 год.)	141
Обробка масивів та рядкових змінних	141
ВИКОРИСТАНА ЛІТЕРАТУРА	144

ПРОГРАМА КУРСУ "МЕТОДИКА НАВЧАННЯ ІНФОРМАТИКИ"

Мета курсу "Методика навчання інформатики" - сформувати у студентів знання, вміння і навички, необхідні для навчання інформатики, використання засобів сучасної інформаційної технології при викладанні інших предметів, для управління навчальним процесом, при його підготовці, супроводі, аналізові, коригуванні, для формування через предмет інформатики та застосування засобів сучасної інформаційної технології в навчальному процесі елементів інформаційної і загальної культури учнів середніх шкіл і СПТУ, інтенсифікації і гуманізації навчального процесу, інтеграції навчальних предметів і індивідуалізації навчання, активізації пізнавальної діяльності дітей, повного розкриття творчого потенціалу вчителя і учня, створення умов для задоволення запитів і розвитку індивідуальних нахилів і здібностей дитини, формування матеріалістичного світогляду, свідомого ставлення до навколишнього світу, розширення і поглиблення теоретичної підготовки, фундаментальних знань і надання результатам навчання практично-значимого характеру.

Для досягнення мети курсу "Методика навчання інформатики" повинен розв'язувати такі **завдання**:

- розкрити значення основ інформаційної культури в загальній і професійній освіті людини, вплив засобів сучасної інформаційної технології на науково-технічний і соціально-економічний розвиток суспільства, з'ясувати психолого-педагогічні аспекти засвоєння предмета, взаємозв'язки шкільного курсу інформатики а іншими навчальними предметами, науково-технічними досягненнями в галузях різних наук, зокрема інформатики, показати практичну значимість методів і засобів сучасної інформаційної технології, можливості їх застосування до розв'язування найрізноманітніших гуманітарних, технічних і наукових проблем;
- забезпечити ґрунтовне вивчення студентами шкільних програм, підручників, навчальних і методичних посібників з інформатики способів використання в навчальному процесі шкільної комп'ютерної техніки і відповідного програмного забезпечення як загального, так і спеціального призначення, розуміння методичних ідей використання методів і засобів сучасної інформаційної технології в навчальному процесі;
- виховати у майбутніх вчителів творчий підхід до розв'язування проблем викладання інформатики та використання обчислювальної техніки в навчальному процесі, сформувати знання, вміння і навички, необхідні для самостійного аналізу навчального процесу, дослідження різноманітних методичних проблем і психолого-педагогічних ситуацій, розвинути здатність і відчуття необхідності до постійної самоосвіти і самовдосконалення, наукового пошуку шляхів удосконалення процесу навчання основам інформатики, підвищення ефективності використання нової інформаційної технології в навчальному процесі, формування елементів інформаційної культури учнів, активізації їх пізнавальної діяльності, творчої активності, самостійного дослідницького характеру пошуку нових знань. З цієї точки зору важливого значення набуває організація самостійної роботи студентів, їх участь у науково-дослідній роботі кафедр, самостійні дослідження у вигляді рефератів, курсових і дипломних робіт, участь у госпдоговірних науково-дослідних роботах, пропаганда серед населення сучасних засобів і методів збирання, зберігання, опрацювання, передавання, подання, використання інформації, роз'яснення їх впливу на розвиток суспільства;
- сформувати у студентів достатні знання, вміння і навички, необхідні для практичного проведення навчально-виховної роботи в умовах широкого використання нової інформаційної технології в навчальному процесі.

ЗНАННЯ

- значення основ інформаційної культури в загальній і професійній освіті людини, вплив засобів сучасної інформаційної технології на науково-технічний та соціально-економічний розвиток суспільства;
- психолого-педагогічні аспекти засвоєння предмета;
- взаємозв'язки шкільного курсу інформатики з іншими навчальними предметами, науково-технічними досягненнями в галузях різних наук;
- практична значимість методів і засобів сучасної інформаційної технології, можливості їх застосування до розв'язування найрізноманітніших гуманітарних, технічних і наукових проблем;
- структура та зміст шкільних програм, підручників, навчальних і методичних посібників з інформатики;
- способи використання в навчальному процесі шкільної комп'ютерної техніки і відповідного програмного забезпечення як загального, так і спеціального призначення;
- використання методів і засобів сучасної інформаційної технології в навчальному процесі;

УМІННЯ

- самостійно проводити аналіз навчального процесу;
- досліджувати різноманітні методичні проблеми і психолого-педагогічні ситуації;
- проводити науковий пошук шляхів удосконалення процесу навчання основам інформатики;
- підвищувати ефективність використання нової інформаційної технології в навчальному процесі;
- вільно володіти програмними засобами загального та спеціального призначення;
- вільно володіти навчальною алгоритмічною мовою та мовою програмування BASIC.

ІНСТРУКТИВНО-МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ЩОДО ВИВЧЕННЯ ІНФОРМАТИКИ У НОВОМУ 2006/2007 НАВЧАЛЬНОМУ РОЦІ

Шкільний курс інформатики в загальноосвітніх навчальних закладах вивчатиметься за програмами, надрукованими у збірнику «Програми для загальноосвітніх навчальних закладів. Навчальні програми для профільного навчання. Програми факультативів, спецкурсів, пропедевтичних курсів, гуртків. Інформатика», видавництво «Прем'єр», Запоріжжя, 2003 р., та новою «Навчальною програмою з інформатики для 8—11 класів загальноосвітніх навчальних закладів універсального та фізико-математичного профілів», надрукованою у науково-методичному журналі «Комп'ютер у школі та сім'ї», №1, 2005 р.

Розподіл годин на вивчення окремих розділів, кількість тематичних оцінювань, передбачених навчальними програмами для загальноосвітніх класів та методичні рекомендації щодо оцінювання навчальних досягнень учнів видруковано в Інформаційному збірнику МОН №№ 13—14, 2002 р., №№ 5, 14—15, 2003 р., №№ 13—14, 2004, 2005 рр., у науково-методичному журналі «Інформатика та інформаційні технології в навчальних закладах», №1, 2006 р.

Орієнтовні критерії оцінювання навчальних досягнень учнів з інформатики з кожної теми курсу, методичні рекомендації для проведення державної підсумкової атестації, видруковані у науково-методичному журналі «Інформатика та інформаційні технології в навчальних закладах», №1, 2006 р.

Продовжується експеримент щодо впровадження інтегрованого навчального курсу «Сходинки до інформатики» у 2—4 класах та курсу «Інформатика» у 5 класах загальноосвітніх навчальних закладах за рахунок варіативної складової навчальних планів (із розрахунку 1 година на тиждень).

Основними завданнями курсу є: формування у молодших школярів первинних навичок роботи за комп'ютером, розуміння ними сутності застосування комп'ютера та інформаційних технологій і загальний розвиток дитини.

Враховуючи віковий, психологічний та фізіологічний стан і розумовий розвиток учня початкової школи, були створені нові сучасні підручники для 2—4 класів. Підручники «Сходинки до інформатики» ознайомлюють учнів із основними поняттями інформатики, залучають до практичної діяльності на комп'ютері, підтримують вивчення основних предметів (українська мова, англійська мова, природознавство, математика), розвивають пам'ять, просторову уяву, логічне мислення та творчі здібності.

Кожний урок у підручниках для 2—4 класів поданий на двох сторінках, тобто на розвороті підручника. Якщо учень пропустив урок, він може вдома самостійно опрацювати матеріал за підручником.

При вивченні курсу передбачено кілька напрямків навчальної та розвивальної діяльності учнів: пізнавальний, прикладний, алгоритмічний та розвивальний.

Опановуючи курс, учні:

- засвоюють відомості про призначення комп'ютера, можливості його використання, складові частини, основні принципи роботи;
- ознайомлюються з властивостями інформації, інформаційними процесами у навколишньому світі;
- здобувають навички роботи з клавіатурою, пошуку та запуску потрібних програм, підготовки та редагування текстів у текстовому редакторі, складання простих мелодій у середовищі музичного редактора, створення малюнків у графічному редакторі та інше;
- засвоюють поняття алгоритму, розрізняють їх основні види, вчать складати і записувати прості алгоритми для виконавців, відшукувати та застосовувати алгоритми у своїй практичній та навчальній діяльності;
- розвивають свої творчі здібності та логічне мислення шляхом виконання різноманітних творчих завдань.

Підтримка, корекція і пропедевтика знань, умінь і навичок з основних предметів досягається шляхом роботи з різноманітними навчальними, навчально-контролюючими та пропедевтичними програмами з української, російської, англійської мов, математики, довідки, античної літератури.

Структура уроків курсу «Сходинки до інформатики» відрізняється від традиційних уроків у молодшій школі і складається з трьох частин: теоретичної, практичної та розвивальної.

Теоретична частина може проводитись у формі бесіди, гри, обговорення ситуацій або призначена для підтримки вивчення навчальних предметів, повторення і закріплення матеріалу. Друга частина — це практична робота учнів за комп'ютерами. Третя частина уроку — це виконання учнями завдань з логічним навантаженням для розвитку пам'яті, кмітливості, ерудиції. Усі частини уроку відображені у підручнику.

Для забезпечення практичної частини уроків інформатики розроблений відповідний програмно-методичний комплекс «Сходинки до інформатики» для 2—5 класів.

Більша частина програм з комплексу «Сходинки до інформатики» сприяють формуванню міжпредметних зв'язків, загальноосвітніх навичок, містять матеріал інших загальноосвітніх предметів. Матеріал програмного комплексу відповідає віковим особливостям учнів, подається в доступній ігровій формі, зацікавлює дітей.

Більша частина навчальних програм мають тренувальний характер та орієнтовані на застосування на етапі формування навичок. Деякі з програм є демонстраційними та можуть бути використані вчителем під час пояснення нового матеріалу або на уроках узагальнення знань.

У 2006/2007 навчальному році експеримент щодо впровадження курсу «Інформатика» для 6 класу буде продовжено у загальноосвітніх навчальних закладах м. Києва.

Обов'язковими умовами впровадження даного курсу є належний матеріально-технічний стан комп'ютерного парку, наявність необхідного програмного забезпечення, дотримання санітарно-гігієнічних вимог та належний рівень підготовленості вчителів. Слід зазначити, що принциповим моментом застосування комп'ютера у початковій школі є те, що відповідний предмет має викладати певним чином підготовлений вчитель початкових класів.

Програму курсів «Сходинки до інформатики» для 2—4 класів та «Інформатика» для 5—6 класів буде видруковано у науково-методичному журналі «Інформатика та інформаційні технології в навчальних закладах» (№ 4, 2006 р.); календарне планування вищезазначених курсів подано у *додатку №1*.

У *вечірніх (змінних)* загальноосвітніх навчальних закладах вивчення курсу інформатики передбачається у 10—12 класах за новою програмою (укладач Литвинова С. Г., Проценко Т. Г.) із розрахунку 1 година на тиждень.

Програма складена з урахуванням того, що у вечірній (змінній) середній школі навчаються учні, які мають деякі прогалини в знаннях, що обумовлені перервою в навчанні, вадами здоров'я, життєвими обставинами тощо.

Програма націлена на формування мотивації до навчання, на розвиток учня як особистості для самостійного набуття знань, їх практичного застосування у повсякденному житті, на розкриття перед учнями можливостей комп'ютера як інструмента пізнання, необхідного в сучасних умовах.

Враховуючи специфіку навчального закладу, доцільно використовувати методи і форми навчання, що забезпечать не тільки засвоєння програмового матеріалу, виховання інформаційної культури учнів, а й розвиток розумових здібностей та якостей особистості (мислення, уваги, пам'яті, пізнавальної діяльності тощо).

Найбільш ефективним для учнів вечірньої школи є проблемно-пошуковий метод навчання. Сутність цього методу полягає в проблемному викладанні знань, пошуковій діяльності учнів (постановка проблемного питання, розв'язання проблеми і проблемних завдань); евристична бесіда, самостійна дослідницька робота учнів на здобуття та розширення знань.

Розподіл годин на вивчення окремих розділів, передбачених навчальною програмою може бути таким:

№ з/п	Тема	Кількість годин
1-й рік навчання (10 клас) – 34 год		
1	Інформація та інформаційні процеси	2
2	Обчислювальна система	5
3	Операційні системи	6
4	Програмні засоби загального призначення 4.1. Графічний редактор 4.2. Текстовий процесор	3 8
5	Програмні засоби навчального призначення (мультимедійні енциклопедії, програми прикладного призначення)	4
6	Глобальна мережа Інтернет. Пошук інформації в Інтернет	4
7	Узагальнення і систематизація знань за рік	2
2-й рік навчання (11 клас) – 34 год		
8	Програмні засоби загального призначення 8.1. Графічний редактор 8.2. Текстовий процесор	3 5
9	Комп'ютерні презентації. Засоби створення комп'ютерних презентацій	7
10	Табличний процесор	7
11	Програмні засоби навчального призначення (програми для підтримки навчання математики, географії, фізики, мов, біології, історії та ін.)	4
12	Електронна пошта. Форуми. Поняття про інтерактивне спілкування в Інтернеті	6
13	Узагальнення і систематизація знань за рік	2
3-й рік навчання (12 клас) – 34 год		
14	Підготовка публікацій за допомогою комп'ютера	7
15	Електронні таблиці. Табличний процесор	7
16	Бази даних. Системи управління базами даних. Експертні системи	10
17	Розширений пошук відомостей в Інтернеті. Оцінювання інформаційних ресурсів Інтернету. Авторське право та Інтернет. Безпека в Інтернеті	4
18	Програмні засоби навчального призначення (програми для підтримки навчання математики, фізики, історії, хімії та ін.)	4
19	Узагальнення і систематизація знань за рік	2

Реалізація *профільного навчання* інформатики у 10—11 класах забезпечується системою курсів за вибором (за рахунок варіативного компонента), які складаються з невеликих за змістом навчальних модулів, враховують різноманіття інтересів і можливостей учнів даного профілю, які поглиблюють та розширюють основний курс інформатики у відповідності до профілю навчання, а також організацією самостійної творчої роботи учнів через систему індивідуальних завдань, спрямованих на розвинення професійних схильностей учнів, їхнього інтересу до застосування інформаційних технологій.

Наведемо приклади нових навчальних програм курсів за вибором з інформатики.

Метою курсу за вибором «*Основи комп'ютерної графіки*» є формування в учнів знань та умінь, необхідних для ефективної обробки інформації, поданої в графічній формі, а також для використання комп'ютерних зображень у навчальній і професійній діяльності.

Мета курсу досягається через опанування учнями необхідного обсягу теоретичного матеріалу та практичне оволодіння сучасних графічно-інформаційних технологій та комп'ютерних засобів і середовищ створення, обробки й візуалізації растрових і векторних зображень. Однаково важливими є уміння створювати як нові зображення та рисунки, так

і редагувати наявні, перетворювати формати комп'ютерних зображень та їхні колірні моделі, імпортувати належним чином підготовлені графічні зображення в офісні документи, у веб-сторінки, в електронні та поліграфічні видання, в рекламу, розробляти комп'ютерну анімацію.

Курс розрахований на 35 навчальних годин і може викладатися в старших класах загальноосвітніх навчальних закладів будь-якого профілю.

Курс має чітко виражену практичну спрямованість, частка часу, відведеного на викладання теоретичного матеріалу, не повинна перевищувати 30%. Програмою курсу передбачено проведення 18 практичних робіт. Власне, вся робота учнів за комп'ютером полягатиме у фронтально-індивідуальному виконанні вправ і завдань зі створення чи обробки графічних зображень. Теоретичний матеріал доцільно викладати невеликими порціями, на початку заняття, орієнтовно, впродовж 10—15 хвилин, оскільки подані теоретичні відомості, якщо вони одразу не закріплені практичною роботою, засвоїти досить важко.

Слід також звернути увагу на те, що не всі діти мають художній хист і його розвиток є лише непрямым супутнім завданням вивчення даного курсу. Тому вправи зі створення малюнків мають бути підібрані так, щоб їх могли виконати учні, які не мають здібностей до малювання. Наголос має робитися на опануванні технічних прийомів і поясненні особливостей роботи в програмному середовищі.

Мета курсу за вибором *«Основи створення комп'ютерних презентацій»* є подвійною: насамперед курс має розвинути в учнів уміння логічно, доступно й ефектно подавати інформацію, структурувати її, застосовувати різні способи її зображення, а також уміння викладати матеріал і зацікавлювати аудиторію. Крім того, учні мають оволодіти сучасними програмними засобами, необхідними для ефективного представлення інформації.

Обсяг курсу становить 17 навчальних годин, з яких принаймні 2 години слід відвести для захисту курсових проектів. Курс розрахований на викладання в загальноосвітніх навчальних закладах будь-якого профілю і може доповнювати тему *«Основи створення комп'ютерних презентацій»* з базового курсу інформатики. Оптимальним є викладання курсу в 10 класі, проте він може бути проведений і пізніше.

Курс має практичну спрямованість, а отже, особливу увагу слід приділяти роботі учнів за комп'ютером. Для навчання роботі з програмою PowerPoint учителям доцільно запропонувати цілісний проект, виконуючи який учні поступово досліджуватимуть та опановуватимуть усі інструментальні можливості програмного продукту. Презентацію, створену в середовищі PowerPoint, згодом можна вдало доповнити за допомогою Microsoft Producer. Що стосується Flash-презентацій, то розробка окремого проекту, у якому органічно розкривався б широкий спектр можливостей програми Macromedia Flash, видається проблематичною. Тому доцільно для кожного заняття формувати окреме завдання.

Слід підкреслити, що навчання має завершуватися виконанням учнями курсового проекту, тобто створенням тематичної презентації, яку учень-автор, або 2—3 автори демонструватимуть своїм однокласникам. Найвірогідніше, проект буде виконуватись учнями протягом усього терміну навчання, замінюючи в такий спосіб домашні завдання. Вдалим рішенням є також виконання і захист двох курсових проектів — одного з технологій Microsoft, іншого з Flash-технологій.

Метою курсу за вибором *«Основи Інтернету»* є формування важливої складової інформаційної культури учнів — знань та вмінь, необхідних для повноцінного та ефективного використання світового інформаційного простору. Завданнями курсу є розвинути в учнів уміння знаходити й систематизувати необхідну інформацію, спілкуватися за допомогою електронних засобів обміну інформацією, використовувати й самостійно освоювати сучасні програмні засоби, формування в учнів теоретичної бази, необхідної для подальшого самостійного освоєння Інтернет-простору, та зрештою, розширення світогляду учнів.

Мета курсу досягається насамперед через практичне оволодіння учнями навичками роботи у світовому інформаційному просторі, використання сучасних програмних засобів роботи в Інтернеті, виконання завдань, що вимагають активного спілкування за допомогою електронних засобів обміну інформацією.

Курс розраховано на 17 годин і може викладатися протягом одного півріччя, по одній годині на тиждень, або протягом 8—9 тижнів, по дві години на тиждень. Особливістю курсу є те, що він вимагає наявності швидкісного інтернет-з'єднання (від 64Кбіт/с, якщо використовується 10—15 учнівських комп'ютерів), але може бути успішно проведений і за наявності повільного з'єднання в режимі віддаленого доступу — в цьому випадку вчителю слід попередньо завантажувати деякі Інтернет-ресурси або ж організовувати роботу учнів у групах так, щоби в онлайн-режимі працювало не більше трьох учнів водночас. У разі відсутності швидкісного з'єднання робота за програмою курсу може бути подовжена на 3—4 години. Крім того, окремі питання можуть вивчатися лише в режимі ознайомлення.

Програмою курсу передбачено проведення 9 практичних робіт, що є найважливішою складовою курсу.

Мета курсу за вибором *«Основи веб-дизайну»* полягає в тому, щоб навчити дітей будувати привабливі, цікаві та інформативні сайти з елементами інтерактивності, а також розміщувати їх в Інтернеті. Завданнями курсу є формування в учнів розуміння принципів організації веб-ресурсів, вироблення вмінь і навичок роботи з сучасними програмними засобами, призначеними для розробки веб-сторінок, а також виховання культури оформлення сайтів й уміння грамотно структурувати інформацію.

Курс розрахований на 35 годин і може викладатися в загальноосвітніх навчальних закладах будь-якого профілю після вивчення в межах базового курсу інформатики основних понять Інтернету та засад комп'ютерної графіки.

Курс містить дві основні складові: технологічну та дизайнерську. Технологічна складова полягає у вивченні мов програмування, програмних засобів і принципів роботи технологій, пов'язаних з розробкою і функціонуванням веб-ресурсів. До дизайнерської складової відноситься проектування інтерфейсу користувача, правил структуризації вмісту веб-сторінок, їх графічного та стилістичного оформлення. Курс побудований так, що технологічна складова вивчається спочатку, дизайнерська — потім. Це пояснюється особливостями веб-дизайну: хоча він і є різновидом художнього дизайну взагалі, але вимагає від дизайнера дотримання певних стратегій і обмежень, які неможливо усвідомити, не розуміючи технологічної складової.

Учні, які повністю опанують курс, ознайомляться з трьома підходами до створення веб-сайтів: автоматизоване створення веб-сайту на безкоштовному сервері, створення сайту в середовищі графічного редактора сайтів та створення сайту мовою HTML. Ця особливість навчального матеріалу дозволяє легко варіювати його обсяги та зміст. Наприклад, у школах гуманітарного або спортивного профілю може бути доцільним обмежитися автоматизованим створенням веб-сайту, у школах природничого, фізико-математичного та художньо-естетичного профілів з матеріалу курсу можна виключити мови HTML та Javascript, а в освітніх закладах інформаційно-технологічного профілю курс можна викладати у повному обсязі.

Курс має практичну спрямованість, на виконання учнями практичних робіт слід відвести не менше 2/3 навчального часу, а завершуватися навчання має виконанням індивідуального чи колективного проекту з розробки тематичного веб-сайту. Вдалим рішенням видається виконання проекту протягом проходження останньої теми, присвяченої дизайнерській складовій курсу. На прикладі сайту, що розробляється в межах цього проекту, можна відпрацьовувати теорію та практику дизайну.

Метою курсу за вибором *«Основи комп'ютерної безпеки»* є формування в учнів важливої складової інформаційної культури — знань та умінь, необхідних для кваліфікованого використання сучасних технологій, стандартів, протоколів та засобів комп'ютерної безпеки. Завданнями курсу є формування в учнів теоретичної бази, необхідної для безпечної роботи з комп'ютером, розвинути умінь використовувати й самостійно освоювати сучасні програмні та технічні засоби захисту інформації, а також надати практичні рекомендації та іншу корисну інформацію, необхідну для того, щоб гарантувати психологічну, моральну та фізичну безпеку дітей під час роботи за комп'ютером.

Вивчення курсу планується протягом одного півріччя, по одній годині на тиждень. Особливістю курсу є те, що він вимагає наявності інтернет-з'єднання, а також наявності спеціального програмного забезпечення, яке вчителю слід попередньо встановити на всіх комп'ютерах учнів. Окремі питання курсу можуть вивчатися лише в режимі ознайомлення без комп'ютера.

Обсяг курсу становить 17 годин і може викладатися в середніх навчальних закладах будь-якого профілю в 10 або 11 класах. Для успішного навчання за тематикою курсу учні повинні мати стійкі навички роботи з прикладними програмами в середовищі Windows. Після вивчення даного курсу в учнів повинен бути сформований необхідний мінімум знань, умінь і навичок, завдяки яким можна успішно використовувати технології і засоби захисту інформації, що зберігається на комп'ютері, а також технології захисту під час доступу до мережі Інтернет.

Курс має практичну спрямованість, оскільки заплановано проведення 9 практичних робіт, які є найважливішою складовою курсу. Слід також зазначити, що для виконання практичних завдань має відводитися не менше половини загального навчального часу.

Метою курсу за вибором *«Основи візуального програмування»* є навчання основам сучасного програмування. У курсі застосована методика, що є для української школи новою, — навчання програмування на прикладах розробки Windows-застосунків. Базовою мовою програмування обрано Visual Basic — найбільш простий і водночас потужний засіб розробки програм. Від учнів не вимагається попереднього знання структурного програмування чи основ алгоритмізації. Перехід від класичної методики до Windows-програмування не призводить до зниження рівня теоретичної підготовки учнів, адже всім базовим концепціям програмування, таким як змінні, типи даних, оператори, алгоритмічні структури, підпрограми й масиви приділяється увага не менша, ніж того вимагає програма базового курсу інформатики.

Курс допоможе ефективно розв'язати дві серйозні проблеми шкільного програмування, пов'язані з нерозумінням учнями сутності предмета та з їхньою низькою мотивацією до навчання. Це досягається завдяки використанню можливостей візуальних засобів сучасних середовищ програмування, які роблять розробку Windows-застосунків простішою й цікавішою, ніж створення класичних алгоритмів і програм. Розроблені учнями програми мають привабливий інтерфейс, принципи їхньої роботи повністю узгоджуються з принципами функціонування всього сучасного програмного забезпечення.

Як і для будь-якого навчання програмуванню, запорукою успішності учня є розв'язання ним великої кількості спеціально підібраних задач. Проте задачі, що пропонуються за тематикою даного курсу, мають формувати в учнів не лише умінь складати й програмувати алгоритми, а й розуміння принципів обробки інформації, специфічних для Windows-застосунків.

Не всі програми учні мають розробляти від початку до кінця, вдалим підходом є створення учителем або використання готових розробок-шаблонів, які учні мають доповнювати фрагментами коду.

Курс розраховано на вивчення протягом одного навчального року, по одній годині на тиждень, навчання за програмою курсу може проводитися також протягом одного півріччя, по дві години на тиждень. Матеріал може викладатися у 10 або 11 класах шкіл, ліцеїв та гімназій усіх профілів як у межах базового курсу інформатики, так і доповнюючи його. Слід зауважити, що в навчальних закладах інформаційно-технологічного профілю курс може стати основою для поглибленого вивчення технології програмування і тому його варто проводити якомога раніше, наприклад у 9 класі.

Курс за вибором *«Інформаційні технології проектування»* рекомендується вивчати паралельно з основним курсом інформатики в 10—11 класах. Курс вивчається за рахунок годин варіативної складової навчального плану із розрахунку 2 години на тиждень (всього 70 годин).

Метою вивчення курсу є формування в учнів загального уявлення про автоматизоване проектування, конструювання та підготовку виробництва у різних галузях людської діяльності. При цьому учні опановують певний обсяг знань та вчать виконувати нескладні операції автоматизованого проектування та підготовки виробництва простих параметризованих виробів у середовищі професійної системи автоматизованого проектування (САПР).

Вивчення цього курсу дозволить ознайомити учнів з методами і допоміжними засобами, які застосовуються під час конструювання і технологічної підготовки виробництва. Передбачається опанування учнями основ електронної обробки даних та основ організації даних, зокрема поданих у графічній формі; вивчення алгоритмів і технологій опрацювання графічної і геометричної інформації; основ геометричного моделювання з теорією параметризації; можливостей

застосування електронних засобів для конструювання і технологічної підготовки виробництва. Учні ознайомляться з рекомендаціями щодо впровадження САПР у промислове виробництво та з перспективами їхнього розвитку, який супроводжується комплексним впливом на традиційні методи організації конструювання і технологічної підготовки. Цілеспрямоване вдосконалення техніки САПР тісно пов'язано з систематизацією, стандартизацією та інтенсифікацією її застосування на виробництві.

Курс за вибором «Сучасні офісні інформаційні технології» рекомендується вивчати паралельно з основним курсом інформатики в 10—11 класах. Курс вивчається за рахунок годин варіативної складової навчального плану із розрахунку 1, 2 чи 4 години на тиждень (всього 70 годин).

Метою курсу є підготовка молоді до активного використання інформаційних технологій у сучасному діловодстві й управлінні; надання знань щодо існуючих норм, правил і способів створення різноманітних документів, способів обміну документами; формування умінь та навичок використання програмно-апаратного забезпечення, яке є основою сучасних комп'ютерно-орієнтованих офісних технологій.

Мета курсу досягається насамперед через ознайомлення учнів з основами інформатики, практичне оволодіння ними сучасних інформаційних технологій та комп'ютерних засобів з ціллю їх активного використання у сучасному діловодстві та управлінні.

Програми наведених вище курсів за вибором будуть надруковані у науково-методичному журналі «Інформатика та інформаційні технології в навчальних закладах» (№ 4, 2006 р.). Навчальні посібники для вивчення запропонованих курсів за вибором готуються до друку у видавництві «ВНУ».

Особистісна спрямованість освіти є однією з основних тенденцій розвитку сучасної школи. Суспільству потрібна компетентна особистість, здатна брати активну участь у розвитку сучасного виробництва, економіки, науки, культури. Тому на перший план шкільної освіти сьогодні висувається завдання створення сприятливих умов для виявлення і розвитку здібностей учнів, задоволення їх інтересів і потреб, розвиток навчально-пізнавальної активності та творчої самостійності.

За цих умов стає актуальним використання *нових технологій навчання*. Вчитель інформатики як предметник працює у складних умовах, що пов'язані з постійним оновленням структури і змісту предмета. Необхідність постійного поновлення власних знань фактичного матеріалу приводить до того, що вчителі інформатики недостатньо уваги приділяють саме методиці викладання предмету. Проте саме володіння вчителем інформатики особистісно орієнтованими загальноосвітніми технологіями, такими як розвивальне, проблемне, модульне навчання позитивно впливає на розвиток учнів, підвищує рівень та якість їх знань.

Проблемне навчання — це навчально-пізнавальна діяльність учнів з засвоєння знань і способів діяльності шляхом сприйняття пояснення вчителя в умовах проблемної ситуації, самостійного аналізу проблемних ситуацій, формулювання проблем і їхнього вирішення через висунення пропозицій, гіпотез, їхнє обґрунтування і доведення, а також шляхом перевірки правильності розв'язання.

Технологія проблемного навчання є природною для курсу інформатики. Під час роботи учнів за комп'ютерами проблемні ситуації виникають постійно і задача учителя полягає в тому, щоб організувати пошуково-дослідницьку роботу учнів з їх розв'язання, надати цьому процесу певну технологічність. Прийомами створення проблемних ситуацій на уроках інформатики є такі:

- підведення учнів до виявлення й усвідомлення сутності певної суперечності з пропозицією самостійно знайти спосіб її вирішення;
- зіткнення певних суперечностей у практичній діяльності;
- викладання різних точок зору на одне і те саме питання;
- розглядання явища з різних позицій;
- аналітична діяльність учня;
- постановка конкретних запитань (на узагальнення, обґрунтування, конкретизацію, логіку, міркування);
- постановка проблемних задач із недостатніми або надлишковими вхідними даними, з суперечливими даними, з явно допущеними помилками тощо.

Проектні технології досить широко використовуються на уроках інформатики. Поширення проектного підходу пояснюється можливістю імітувати у навчальному кабінеті діяльність, що реально розвертається у позанавчальному житті.

Застосування проектного навчання можливо не тільки у вигляді великих комплексних проектів, але і під час засвоєння окремих елементів комп'ютерних технологій.

Курсовий проект припускає виконання самостійної творчої роботи учнів протягом певного часу. Успіх методу курсових проектів багато в чому залежить від учителя, його тактовності і контролю за дотриманням порядку виконання курсової роботи.

Робота за методом курсових проектів є цікавою і для вчителів-предметників. Комп'ютер у даному випадку стає не стільки об'єктом, скільки інструментальним засобом проектування. Проектна технологія використовується у багатьох навчальних закладах, зокрема у Дніпропетровському ліцеї інформаційних технологій при Дніпропетровському національному університеті, де учні створюють навчальні проекти під спільним керівництвом вчителя інформатики і вчителів-предметників.

Надзвичайно ефективним у навчальному процесі є використання комунікаційних можливостей всесвітньої мережі, тобто спеціально організованої діяльності, у якій учасники з різних регіонів міста, країни, різних країн світу разом співпрацюють над цікавою для всіх учасників проблемою, обмінюючись за допомогою швидкодійної мережі не тільки результатами своєї діяльності, а й методами дослідження, обговорюючи проміжні результати, спілкуючись з опонентами й однодумцями. Саме так організована навчальна діяльність учнів та вчителів у роботі над телекомунікаційними проектами в деяких навчальних світових мережах.

Можливість для творчості, самовираження — ці складові телекомунікаційних проєктів є суттєвими для створення потрібної мотивації для вчителів використовувати Інтернет у навчальному процесі.

Навчальний *телекомунікаційний проєкт* — це спільна навчально-пізнавальна творча або ігрова діяльність учнів-партнерів, що організована на основі комп'ютерної телекомунікації та має спільну мету — дослідження певної проблеми, погодження методів та способів діяльності, що направлені на досягнення спільного результату діяльності. Особливостями телекомунікаційних проєктів є те, що:

- телекомунікаційні проєкти за своєю суттю завжди міжпредметні, тобто розв'язання проблеми завжди вимагає залучення інтегрованого знання;
- організація телекомунікаційних проєктів вимагає спеціальної досить ретельної підготовки як вчителів, так і учнів;
- телекомунікаційні проєкти дозволяють не тільки передавати учням суму знань, а також вчать їх здобувати ці знання самостійно за допомогою мережі Інтернет; уміти користуватися отриманими знаннями для розв'язування пізнавальних і практичних завдань;
- вчителі і учні у процесі роботи у телекомунікаційному проєкті набувають комунікативних навичок і вмінь.

Виділяють такі типи проєктів: дослідницькі, літературно-творчі, пригодницькі, ігрові, інформаційні, практично-орієнтовані тощо. Цікавим є досвід використання телекомунікаційних проєктів у Чернівецькій гімназії № 2 (учитель інформатики – Спори́ніна Т. Т.), СЗОШ №94 з поглибленим вивченням івриту та інформатики м. Одеси (вчитель Пахомова Г. В.), ліцею № 62 м. Запоріжжя (учитель інформатики Бессонова Н.О.).

Разом з іншими інформаційними технологіями комп'ютерна комунікація може істотно вплинути на формування нового змісту освіти, на зміни організаційних форм і методів навчання. Використовуючи телекомунікаційні технології у навчальному процесі, вчитель інформатики повинен націлювати учнів на:

- пошуково-дослідницьку роботу, яка надає можливість використовувати потужні ресурси Інтернет;
- телекомунікаційну проєктну діяльність, яка надає учасникам проєкту можливість для сумісної навчально-пізнавальної, дослідницької, творчої діяльності при вирішенні загальних проблем;
- участь у телекомунікаційних, відео- та Інтернет-конференціях, яка не тільки надає змогу ознайомитись із найсучаснішими телекомунікаційними технологіями, а й має велике мотиваційне та стимулююче значення у навчанні, надає можливість візуального спілкування та представлення сумісних результатів;
- навчання у віртуальних навчальних закладах.

Глобальна комп'ютерна мережа Інтернет відкриває реальні можливості повсякденного співробітництва вчителів та учнів різних освітніх закладів. Мережа стала посередником спілкування між людьми, посередником, який створив «віртуальні світи» зі спеціальними психологічними властивостями.

Приблизно 4% дітей у віці 12—16 років грають в комп'ютерні ігри 30 годин на тиждень. З 400 учнів 10—11 класів столичних шкіл 80 школярів віддали перевагу спілкуванню через Інтернет, 60 — безпосередньому спілкуванню, решта — в залежності від ситуації.

Аргументуючи свій вибір на користь спілкування через Інтернет, школярі зазначають:

- у мережі спілкування простіше, вільніше, без обмежень;
- у віртуальному просторі можна діяти інкогніто;
- можна легко припинити спілкування;
- можна реалізувати свої фантазії.

Кількість часу, яку діти проводять в онлайні, є джерелом хвилювань багатьох батьків. Спочатку батьки з радістю приймають Інтернет у своїх домах, вважаючи, що він відкриє захоплюючий новий світ навчальних можливостей для їхніх дітей. Проте дуже скоро батьки розуміють, що, замість того, щоб використовувати Інтернет для виконання домашніх завдань і досліджень, діти проводять там години, обмінюючись миттєвими повідомленнями із своїми друзями, граючись в онлайніві ігри або розмовляючи з незна-йомими людьми в чат-кімнатах.

Інтернет є прекрасним інструментом для молодих людей, особливо для тих, хто відчуває труднощі у спілкуванні зі своїми ровесниками. Діти, які гарно володіють комп'ютером, можуть користуватися популярністю в Інтернеті, оскільки зовнішність та атлетичні здібності там не є важливими, і це може допомогти їм набути гарну самооцінку. Тим не менш, надмірне використання комп'ютера може ще більше ізолювати сором'язливих, замкнених дітей від інших видів діяльності, таких як виконання домашньої роботи, заняття спортом, сон або спілкування з іншими дітьми. Батьки та вчителі зазвичай не знають, що існує така проблема, доки вона не проявиться явно й не стане серйозною.

Проблема полягає в тому, що «віртуальна залежність» може привести до соціальної та емоційної ізоляції, інформаційного перевантаження, до індивідуалізму та відчуження. Слід зазначити, що «віртуальна залежність» найчастіше виникає у людей дезадаптованих, тих, у кого з різних причин відсутнє повноцінне реальне спілкування в силу їх заниженої самооцінки, некоммукабельності тощо. На перший погляд здається, що Інтернет-спілкування — це прекрасний вихід для таких людей. Але цей «вихід» лише поглиблює проблему, допомагає уникати її актуалізації та розв'язання в реальній дійсності.

Виділяють такі основні типи так званої Інтернет-залежності:

захоплення віртуальними знайомствами — велика кількість віртуальних друзів у Мережі;

нав'язлива потреба у Мережі — гра в онлайніві азартні ігри, постійна участь в аукціонах;

інформаційне перевантаження — постійні «мандрування» Мережею, пошук інформації в базах даних та пошукових сайтах;

комп'ютерна залежність — нав'язлива потреба в комп'ютерних іграх;

кіберсексуальна залежність — потреба в постійному відвідуванні порно-сайтів.

Унікальність мережі Інтернет полягає в її відкритості, доступності та відсутності прав власності. Тому практично в усіх її проявах відсутні регулювання, цензура та інші методи контролю інформації. Завдяки цьому відкриваються необмежені можливості доступу до будь-якого виду інформації.

Під час проведення уроків з використанням мережі Інтернет перед учителем постає дуже важлива проблема: не допустити можливості доступу учнів до сайтів, що містять негативну інформацію. І саме вчителі повинні навчати учнів правильному користуванню Мережею і порадити батькам, яким чином встановити здоровий баланс між використанням Інтернет та іншими видами діяльності.

Інформацію щодо безпеки дітей в Інтернеті можна отримати на сайті: <http://www.microsoft.com/ukraine/athome/security/children>.

Складовою частиною навчального процесу є контроль за навчальною діяльністю учнів, призначений для визначення успішності навчання кожного учня, аналізу отриманих результатів та корекції подальшого процесу навчання. Інструментом підвищення якості освіти разом із реформуванням змісту освіти виступає вдосконалення системи оцінювання, модернізація діагностики навчання учнів. Необхідність забезпечувати контроль і оцінювання не тільки результату, а й процесу навчання сприяє пошуку оперативних та об'єктивних методів контролю знань.

Залежно від функцій, які виконує контроль у навчальному процесі (навчаюча, вимірювання та оцінювання, виховна та розвиваюча, прогностично-методична), виділяють три основні його види: попередній, поточний та підсумковий. Вид контролю визначається метою його проведення.

Нині існує достатня кількість педагогічних програмних засобів призначених для здійснення контролю знань. Найбільш поширеним та перспективним для використання в навчальному процесі є програми-оболонки, що дозволяють створювати тестові завдання та методичний супровід до них, формувати набори запитань і використовувати їх при проведенні контролю знань. Цей клас програм передбачає використання комп'ютера як в процесі підготовки до проведення контролю, так і при його проведенні, а також обробки його результатів.

Педагогічно доцільне та методично грамотне використання педагогічних програмних засобів забезпечує реалізацію основних функцій контролю (див. таблицю), включаючи своєчасне коригування знань, умінь та навичок учнів та ефективне управління їх навчально-пізнавальною діяльністю.

Вид контролю	Функції контролю	Мета використання ППЗ	ППЗ, що використовуються	Функції ППЗ
Попередній	прогностично-методична; вимірювання та оцінювання; виховна	виявлення індивідуальних психологічних особливостей учнів; виявлення рівня підготовки учнів з теми, що буде вивчатись	тестуючі програми; автоматизовані системи контролю знань	діагностуюча; контролююча
Поточний	навчаюча; вимірювання та оцінювання; прогностично-методична; виховна	реалізація можливостей самоконтролю учнів у процесі навчання; оцінювання та корегування результатів навчально-пізнавальної діяльності учнів	контрольно-навчаючі програми; тренажери; автоматизовані системи контролю знань	інформаційна; організуюча; діагностуюча; коригуюча; контролююча
Підсумковий	вимірювання та оцінювання; прогностично-методична; виховна	проведення підсумкового контролю після закінчення вивчення розділу (теми); проведення комплексного контролю після закінчення вивчення курсу	автоматизовані системи контролю знань; тестуючі програми	контролююча; діагностуюча

Одним із напрямків удосконалення контролю навчальних досягнень учнів є тестові технології.

До переваг комп'ютерної форми тестування варто віднести:

- об'єктивність оцінювання;
- зручність фіксації, збереження та представлення результатів тестування;
- можливість автоматизованого опрацювання результатів, ведення баз даних та статистичного аналізу;
- можливість індивідуалізації процесу навчання;
- використання графічних, динамічних, інтерактивних та інших можливостей подання тестових завдань.

Для оцінювання навчальних досягнень учнів можна запропонувати контрольно-діагностичну систему TEST-2005 (авт. Є. Шестопапов), до складу якої входять: діагностуюча оболонка, редактор тестів, конвертер, папки з тестами.

Одна з технологій, що застосовуються для оцінювання якості освіти — це метод оцінювання за допомогою *портфоліо*. Портфоліо учня — файлова папка, що містить електронну версію аналітичного реферату, електронну версію статті до газети або журналу, Web-сторінку проекту, комп'ютерну презентацію проекту.

Вдало поєднує у своїй роботі проектні та інформаційні технології з технологією портфоліо Сідельник С. О., учитель інформатики ЗОШ І—ІІІ ступенів № 61 м. Донецька. Матюхін О. В., вчитель гуманітарно-правового ліцею м. Черкас, використовує мережеву технологію тестового оцінювання навчальних досягнень учнів. Цікавим є досвід Іштукіна В.В.,

вчителя Прилуцької ЗОШ І—ІІІ ступенів № 6 Чернігівської області, який у своїй роботі використовує ігрові технології навчання зі спонукально-заохочувальним контролем рівня навченості.

Оцінювання навчальних досягнень учнів з інформатики є досить специфічним питанням з огляду на значну прикладну складову змісту навчання. Тому вчителям інформатики слід звернути особливу увагу на проведення і оцінювання практичних робіт, під час виконання яких учень розширює і поглиблює свої теоретичні знання, вчиться використовувати отримані знання на практиці, застосовувати вміння і навички для розв'язування нетрадиційних завдань, у тому числі прикладного змісту. Звертаємо увагу на те, що виставлення балів до класного журналу за кожну практичну роботу кожному учню *не є обов'язковим*.

Інтенсивне застосування комп'ютерних засобів у навчально-виховному процесі вимагає уважного й відповідального розгляду питань забезпечення безпеки учнів та вчителів, а також розробки відповідних рекомендацій, виконання яких дозволить захистити фізичне й психічне здоров'я людей від негативного впливу технічних і програмних засобів інформаційних технологій.

Загалом на функціональний стан молодшої людини та на її здоров'я під час навчання в комп'ютерному класі впливає комплекс об'єктивних і суб'єктивних чинників, зокрема, зміст і обсяг навчальної інформації, інтенсивність і тривалість роботи за ПК, складність навчального предмета, якість і досконалість використовуваних програмних продуктів, їхні ергономічні, педагогічні, психогігієнічні властивості та рівень «дружності» інтерфейсу.

Окрім того, об'єктивними, гігієнічно значимими також вважають чинники внутрішнього середовища навчального приміщення, які виникають під час роботи комп'ютерних комплексів – показники мікроклімату, освітленість, яскравість, контрастність і колір зображення на екрані дисплея, іонізуюче та неіонізуюче опромінення, шум тощо.

При облаштуванні й обладнанні комп'ютерних кабінетів, нормуванні тривалості роботи, зокрема учнів і вчителів, необхідно неухильно дотримуватися санітарних, ергономічних, гігієнічних норм та проводити певні фізкультурно-оздоровчі заходи. Для того щоб зняти напруження очей, залежно від рівня втоми очей, рекомендуються певні вправи, які виконуються індивідуально кожним користувачем ПК. Варіанти таких вправ подано у додатку № 2.

Виконання профілактично-оздоровчих заходів дозволить усім працюючим за комп'ютерами значно зменшити їх вплив на здоров'я, фізичний стан та психіку людини. При цьому слід зазначити, що для загальноосвітніх навчальних закладів вже розроблені й введені в дію Державні санітарні правила та норми влаштування і обладнання кабінетів комп'ютерної техніки та режим праці учнів.

Ще раз наголошуємо, що учителі інформатики повинні слідкувати за дотриманням правил безпеки під час навчання в кабінетах інформатики (наказ Держнаглядохоронпраці України від 16.03.2004 р. № 81).

Додаток № 1

Календарне планування курсів «Сходинки до інформатики» (2–4 класи)

та «Інформатика. 5 клас»

2 клас

(1 година на тиждень)

1. Техніка безпеки при роботі з комп'ютером і правила поведінки в комп'ютерному класі. Перегляд комп'ютерного мультфільму. 1 год.
2. Обговорення можливостей і демонстрація режимів роботи комп'ютера. Складові частини комп'ютера. Формування навичок роботи з мишею. 3 год.
3. Розвиток логічного мислення і математичних здібностей. Істинні та хибні висловлення. Закріплення навичок роботи з мишею. Розвиток просторової уяви. Комп'ютерна підтримка уроків математики. 3 год.
4. Короткі історичні відомості про обчислювальні прилади та з історії створення ЕОМ. Комп'ютерна підтримка навчальних предметів. 3 год.
5. Робочий стіл. Піктограми. 1 год.
6. Інформація. Як людина сприймає інформацію. Що можна робити з інформацією. Інформаційні процеси. Робота в графічному редакторі Paint. 3 год.
7. Підсумковий урок за I семестр. 1 год.
8. Що можна робити з інформацією. Інформаційні процеси. Робота в графічному редакторі Paint. 5 год.
9. Ознайомлення з розміщенням і призначенням клавіш на клавіатурі. Робота з клавіатурним тренажером. 4 год.
10. Створення вітальної листівки з використанням графічного редактора і тексту. Друкування листівки. 1 год.
11. Комп'ютерна підтримка вивчення навчальних предметів (українська мова, англійська мова, математика, ознайомлення з навколишнім світом, ОБЖ). 6 год.
12. Підсумковий урок за рік. 1 год.

3 клас

(1 година на тиждень)

1. Техніка безпеки при роботі з комп'ютером і правила поведінки в комп'ютерному класі. Робота з програмою на розвиток пам'яті. 1 год.
2. Як підготувати комп'ютер до роботи. Поняття про операційну систему. Операційна система Windows. Вікна. Робота з вікнами. Калькулятор. 3 год.
3. Інформація, інформаційні процеси (повторення). 2 год.
4. Місце і способи збереження інформації. Пам'ять. Робота з програмами в підтримку навчальних предметів. 3 год.
5. Передавання інформації. Робота з програмами в підтримку навчальних предметів. 2 год.

6. Перетворення інформації. Кодування. Робота з програмами в підтримку навчальних предметів. Клавіатурний тренажер. 2 год.
7. Обробка інформації. Процесор. Клавіатурний тренажер. 2 год.
8. Підсумковий урок за I семестр. 1 год.
9. Текст. Тема, заголовок, будова тексту, абзац. Типи текстів. Робота в текстовому редакторі WordPad. Запис тексту на диск. 5 год.
10. Графічний редактор Paint. 2 год.
11. Типи речень за метою висловлення. Спонукальні речення. Поняття команди. 2 год.
12. Команди і виконавці. 2 год.
13. Поняття алгоритму. Складання алгоритмів для виконавців. 4 год.
14. Ханойська вежа. 1 год.
15. Підсумковий урок за рік. 1 год.

4 клас

(1 година на тиждень)

1. Техніка безпеки при роботі з комп'ютером і правила поведінки в комп'ютерному класі. Робота з розвиваючою програмою. 1 год.
2. Інтернет. Всесвітня павутина. Робота з Internet Explorer. 4 год.
3. Електронна пошта. Надсилання і одержання листа. 1 год.
4. Алгоритми (повторення). Робота з виконавцями. 2 год.
5. Лінійні алгоритми. Блок-схеми лінійних алгоритмів. 2 год.
6. Розгалуження. Алгоритми з розгалуженням. Блок-схеми алгоритмів з розгалуженням. Складання алгоритмів з розгалуженням для виконавців. 2 год.
7. Цикли. Блок-схеми алгоритмів з циклами. Складання алгоритмів з циклами для виконавців. 2 год.
8. Підсумковий урок за I семестр. 1 год.
9. Істинні та хибні висловлення. Слова-заперечення. Заперечення висловлень. Логічне слідування. Складання алгоритмів з циклами для виконавців. 3 год.
10. Диски, файли, папки. Програма «Мій комп'ютер». 2 год.
11. Тести про тексти та про їх редагування в текстовому редакторі. Повторення і нові прийоми форматування і редагування текстів. Завантаження тексту з диска. Робота в текстовому редакторі WordPad. Запис документа на диск. 3 год.
12. Введення і форматування прозових і віршованих текстів. 2 год.
13. Робота з деформованим текстом. Копіювання і переміщення фрагментів тексту. 1 год.
14. Музичні можливості комп'ютера. Робота з музичним редактором. 4 год.
15. Стандартні програвачі мелодій. 2 год.
16. Підсумковий урок за рік. 1 год.

5 клас

(1 година на тиждень)

1. Техніка безпеки при роботі з комп'ютером і правила поведінки в комп'ютерному класі. Інформація. Як людина сприймає інформацію. Інформаційні процеси. Основні блоки комп'ютера, їх призначення. Сфери застосування сучасного комп'ютера. Включення комп'ютера, запуск програми. Закінчення сеансу роботи. Вимкнення комп'ютера. (повторення). 1 год.
2. Операційна система **Windows**. Вікна, їх елементи. Зміна розмірів і розташування вікон. Смуги прокрутки. Переключення між вікнами (повторення). 2 год.
3. Вікно **Мій комп'ютер**. Диски, види дисків. Локальні і мережеві диски, їх імена. Файли. Ім'я файлу. Папки, ім'я папки. Дерево папок. Навігація по дисках і папках (зміна поточного диска (папки), кнопки «Вгору», «Назад», «Вперед»). Зміна вигляду і порядку виведення значків у папці. 1 год.
4. Створення і знищення папок, зміна імені папки за допомогою контекстного меню. Скасування останньої операції. 1 год.
5. Копіювання і переміщення файлів і папок. Самостійна робота. 2 год.
6. Налаштування екрана. 1 год.
7. Робота з логічними програмами. 1 год.
8. Тематичне оцінювання. 1 год.
9. Графічний редактор **Paint**. Інструменти і палітра (повторення) 1 год.
10. Робота з фрагментами малюнка. 1 год.
11. Перетворення малюнків. 1 год.
12. Симетричні фігури. 1 год.
13. Тематичне оцінювання. 1 год.
14. Алгоритми і виконавці. Способи запису алгоритму. Лінійні алгоритми і алгоритми з розгалуженням (повторення). 2 год.
15. Виконавець **Черепашка**, його система команд. Робота з виконавцем у безпосередньому режимі з використанням кутів 90° і 180°. Команда **Color**. 1 год.
16. Кути повороту. 1 год.
17. Складання і виконання лінійних алгоритмів для виконавця **Черепашка**. 1 год.

18. Алгоритми з циклами. Складання і виконання алгоритмів з циклами для виконавця **Черепашка**. Самостійна робота. 3 год.
19. Тематичне оцінювання. 1 год.
20. Текстовий редактор **WordPad**. Ведення, редагування і форматування тексту (повторення). 2 год.
21. Робота з фрагментами тексту. Самостійна робота. 1 год.
22. Створення нового документа. Створення документа з малюнками. 2 год.
23. Тематичне оцінювання. 1 год.
24. Робота з розвивальними програмами. 2 год.
25. Резервний час. 3 год.

Додаток № 2

Вправи для очей

Варіант 1. Заплющити очі на кілька секунд, сильно напружуючи м'язи очей, потім розплющити їх, розслабляючи м'язи очей. Дихання ритмічне. Повторити 4—5 разів.

Подивитись на перенісся і затримати подих на час кількох повних циклів дихання. До втоми очі доводити не можна. Потім перевести погляд удалечінь (подивитись у вікно). Повторити 4—5 разів.

Не повертаючи голови, подивитись направо і зафіксувати погляд на кілька секунд, потім подивитись вдалечінь прямо. Аналогічно виконати вправу з фіксацією наліво, угору, дотолу. Повторити 4—5 разів.

Часто заплющувати і розплющувати очі (кліпати) протягом 20—30 секунд.

Варіант 2. Дуже повільно переводити погляд: вдихаючи, дивитись вправо, потім угору, видихаючи — вліво і дотолу. Потім спрямувати погляд удалечінь. Повторити вправу, спрямовуючи погляд у протилежний бік. Виконати вправу 4—5 разів в один бік і стільки ж — в інший.

Для покращення фокусування ока подивитись на кінчик носа, потім одразу перевести погляд у далечінь. В обох випадках погляд затримати на кілька секунд. Повторити вправу 4—5 разів.

Заплющити очі на 3—5 секунд, а потім розплющити їх на такий же час. Повторити вправу 4—5 разів.

Не повертаючи голови, переносити погляд угору-прямо-дотолу, потім — вліво-прямо-вправо. Аналогічно виконувати рухи очей по діагоналі в один та інший бік.

Повторити комплекс вправ 4—5 разів.

Варіант 3. Подивитись вправо, не повертаючи голови, і зафіксувати погляд на кілька секунд. Потім — вліво на кілька секунд і подивитись прямо. Спрямувати погляд угору на кілька секунд, затим дотолу, розслабивши м'язи очей, насамкінець — прямо перед собою. Повторити вправу 4—5 разів.

Подивитись кілька секунд на кінчик пальця, віддаленого на 30 см від очей, потім перенести погляд удалечінь і затримати погляд на 3—5 секунд. Вправу повторити 4—6 разів.

Заплющити очі на 5—10 секунд, розслабивши м'язи очей, і розплющити їх на такий же час. Повторити вправу 4—5 разів.

Зробити кілька колових рухів (4—5) очима в один бік, а потім в другий. Темп середній. Перенести погляд прямо перед собою на 5—7 секунд. Повторити вправу 3—4 рази.

Покліпати кілька разів очима, не напружуючи м'язи очей.

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ВИВЧЕННЯ КУРСУ “МЕТОДИКА НАВЧАННЯ ІНФОРМАТИКИ”

Метою шкільного курсу інформатики є формування теоретичної бази знань учнів з основ інформатики та практичних навичок використання засобів сучасних інформаційних технологій у повсякденній практичній, зокрема навчально-пізнавальній, діяльності учнів.

До теоретичної бази знань відносяться: уявлення про інформацію, її властивості, інформаційні процеси та інформаційні системи, загальні принципи розв'язування задач за допомогою комп'ютера при використанні програм загального та прикладного забезпечення, формулювання проблем і постановка задач, побудова відповідних інформаційних (зокрема, математичних) моделей, основи алгоритмізації і програмування, принципи будови та дії комп'ютера, уявлення про можливості глобальної мережі Інтернет, пошук потрібної інформації.

До практичних навичок відносяться: навички роботи з пристроями введення-виведення інформації, прикладним програмним забезпеченням загального й цільового призначення: редакторами текстів, графічними редакторами, електронними таблицями, системами управління базами даних, інформаційно-пошуковими системами, педагогічними програмними засобами, програмами-броузерами для перегляду гіпертекстових сторінок; програмами для роботи з електронною поштою і телеконференціями; пошук інформації в глобальній мережі Інтернет тощо; навички складання, описування та реалізації деяких алгоритмів і програм з використанням засобів навчальної алгоритмічної мови та реальних мов програмування і операційних систем.

Мета курсу досягається через практичне оволодіння учнями навичками роботи з основними складовими сучасного програмного забезпечення ЕОМ, ознайомлення з функціональним призначенням основних пристроїв комп'ютера та принципами їх будови і дії, основами технології розв'язування задач за допомогою комп'ютера, починаючи від їх постановки й побудови відповідних інформаційних моделей і закінчуючи інтерпретацією результатів, отриманих за допомогою комп'ютера.

Програма курсу розрахована на вивчення інформатики за умов достатнього доступу учнів до комп'ютерів.

Залежно від типу обчислювальної техніки, складу наявного науково-методичного та програмного забезпечення учитель може самостійно добирати методичні шляхи розв'язання освітніх завдань курсу, вносити необхідні корективи в

порядок вивчення тем програми, а також змінювати кількість годин, необхідних для засвоєння програмного матеріалу. Окремі питання програми можуть вивчатися тільки в порядку ознайомлення. Відповідно до обраної методики викладання курсу вчитель може дібрати відповідні навчальні посібники та дидактичне забезпечення, віддаючи перевагу тим чи іншим з них або ж певним чином поєднуючи їх.

Для забезпечення курсу інформатики необхідні такі програмні засоби:

1. Клавіатурний тренажер - для оволодіння клавіатурою ПЕОМ.
2. Операційна система - для засвоєння правил роботи з системою введення-виведення інформації та її зберігання на зовнішніх носіях.
3. Операційна оболонка - для засвоєння правил роботи з файлами, що зберігаються на зовнішніх носіях.
4. Редактор текстів - для засвоєння правил введення, зберігання та редагування текстів на ПЕОМ.
5. Графічний редактор - для засвоєння правил виконання на ПЕОМ малюнків, схем, креслень, анімацій тощо.
6. Електронні таблиці - для засвоєння правил виконання значних за обсягом розрахунків за їх допомогою та ознайомлення з діловою графікою.
7. Системи управління базами даних - для засвоєння правил створення, редагування, впорядкування, пошуку та нагромадження інформації в базах даних.
8. Довідково-інформаційна система.
9. Набір прикладних програм цільового призначення (педагогічні програмні засоби).
10. Набір програм для роботи в глобальній мережі Інтернет - для перегляду Web-сторінок, роботи з електронною поштою та телеконференціями, файловими архівами, пошуковими машинами тощо.
11. Діалоговий інтерпретатор (компілятор) однієї з мов програмування (Basic, Pascal, C, Visual Basic, Delphi тощо) - для ознайомлення з правилами складання, введення, виконання та налагодження програм на ПЕОМ.

При вивченні курсу "Методика навчання інформатики" лекції можуть бути замінені лабораторними роботами. Всі необхідні теоретичні відомості при цьому можуть подаватися в інструкціях до лабораторних робіт і вивчатися студентами самостійно а подальшим уточненням під час виконання та захисту лабораторних робіт або ж подаватися викладачем перед початком виконання лабораторної роботи. При вивченні поняття інформації слід звернути увагу студентів на те, що це поняття не означає, означення цього поняття не існує, як і не існує вичерпної відповіді на запитання про те, що ж таке інформація. Розглядаючи питання про оцінку і вимірювання інформації, слід звернути увагу на суб'єктивний характер оцінки інформації, на взаємозв'язок і взаємоперетворення інформації і шуму, на неоднозначність тлумачення повідомлень залежно від обставин. Досить поширені поняття біт, байт, кілобайт і т.д. слід застосовувати до характеристики ємності запам'ятовуючих пристроїв, полів пам'яті оперативного запам'ятовуючого пристрою, ємності регістрів, магнітних дисків тощо і не трактувати їх як одиниці вимірювання інформації, оскільки в інформатиці можуть бути різні підходи до оцінки і вимірювання інформації. До того ж в теорії інформації в поняття інформації вкладається зовсім інший зміст, ніж в інформатиці. Розглядаючи зв'язок повідомлення S і інформації I , яка при цьому одержується за певним правилом інтерпретації, доцільно ввести поняття алгоритму, як деяке поопераційне задане правило інтерпретації повідомлення, не заглиблюючись поки що у правила конструювання алгоритмів, але відмітивши всі їх властивості, які дають можливість чітко визначити, є деяка послідовність вказівок алгоритмом, чи ні. На цю обставину слід звернути особливу увагу, оскільки досить поширеними є помилки, коли алгоритмами називають послідовність вказівок, які насправді не можна відносити до алгоритмів.

Пристаюючи до формування вмінь і навичок користувача опрацювання інформації за допомогою ЕОМ, доцільно ознайомити студентів із засобами обчислення значень функцій і виразів, найпростішими вказівками введення і опрацювання даних, редагування тексту, виробити початкові навички роботи з текстовим і графічним редактором, сформулювати поняття про програмне забезпечення загального призначення і прикладне програмне забезпечення, ознайомити із найпоширенішими і найчастіше вживаними прикладними програмами.

Пристаюючи до вивчення основ алгоритмізації, слід при побудові алгоритмів обов'язково спиратися на базові структури алгоритмів і чітко виділяти їх в структурі алгоритму, при необхідності супроводжуючи процес побудови алгоритму за методом послідовної деталізації зверху-вниз відповідним унаочненням (схемами алгоритмів). При цьому слід наголосити на трьох основних базових алгоритмічних структурах (слідування, розгалуження, повторення) і не вводити до схем алгоритмів ніяких інших структур. Немає також потреби дотримуватись при побудові, схем алгоритмів формальних вимог ГОСТу. Слід повторити правила описування алгоритмів навчальною алгоритмічною мовою, викладеною в шкільних посібниках, і обов'язково дотримуватись відповідних правил конструювання алгоритмів, у тому числі і їх схем.

При вивченні неструктурованої мови програмування типу BASIC доцільно розглянути правила побудови структурованих описів програм на основі неструктурованої мови програмування. Це дає змогу будувати описи програм, які мають практично всі властивості описів структурованою мовою типу PASCAL. Такі описи слід супроводжувати відповідними коментарями, особливо при значному об'ємі вказівок у програмі.

При вивченні режиму прямих вказівок мовою BASIC доцільно розглянути приклади програм з циклами і розгалуженнями, описи яких вміщуються в один прямий логічний рядок, що дасть можливість і розглянути приклади нескладних структурованих програм, описаних мовою BASIC.

У лекційному курсі передбачається розкрити мету і завдання викладання інформатики та використання засобів нової інформаційної технології в навчальному процесі середніх шкіл і СПТУ, розкрити структуру, мету, зміст шкільного та факультативних курсів основ інформатики та обчислювальної техніки, - ознайомити майбутніх вчителів з найважливішими питаннями методології інформатики, з метою, змістом, методами, організаційними формами і засобами навчання, орієнтованими на нову інформаційну технологію, а також висвітлити найважливіші теми спеціально: методики викладання основ інформатики. Зміст лекцій необхідно узгоджувати з уже відомими студентам відомостями з філософії, психології,

педагогіки, логіки, інформатики, математики, фізики, математичної логіки, обчислювальної математики, теорії імовірностей і математичної статистики та інших предметів.

Необхідно також знайомити студентів з передовим досвідом вчителів інформатики, роботою передових і експериментальних шкіл, шкіл-лабораторій, педагогічних класів відповідної орієнтації, окремих СПТУ, підприємств а належним чином організованими автоматизованим виробництвом і довідково-інформаційними системами, довідково-інформаційними системами масового обслуговування і т.п.

Послідовність вивчення окремих тем може варіюватися відповідно до термінів проведення педагогічної практики, організаційно-технічного і науково-методичного оснащення навчального процесу тощо. Деякі теми за рішенням кафедри можуть бути запропоновані студентам для самостійного опрацювання.

На семінарських заняттях передбачається вивчення діючих шкільних програм, підручників, навчальних і методичних посібників, типів шкільної комп'ютерної техніки, комп'ютерних програм загального і спеціального призначення, зокрема педагогічних програмних засобів. Крім того, передбачається ознайомлення студентів із структурою діяльності вчителя інформатики та плануванням його роботи, складанням конспектів, аналізом уроків різних типів, організацією самостійної роботи учнів, формами, методами і засобами контролю навчальної діяльності учнів та методикою її оцінювання в умовах широкого використання нової інформаційної технології в навчальному процесі. Передбачається також добір та обговорення матеріалів для факультативних та позакласних занять, проведення ролевих ігор, диспутів, змагань, зустрічей студентських груп типу "Клуб веселих і кмітливих", "Що? Де? Коли?", відвідування і обговорення уроків передових учителів, уроків у школах-лабораторіях, експериментальних школах тощо.

На лабораторних заняттях студенти можуть складати конспекти уроків різних типів, готувати комп'ютерне забезпечення уроків та позакласних заходів, проводити логіко-дидактичний аналіз наявних педагогічних програмних засобів, проводити ділові ігри, вчитися аналізувати уроки, перевіряти виконані учнями за допомогою комп'ютерів завдання, виготовляти фрагменти педагогічних програмних засобів, унаочнення навчального процесу, дидактичні матеріали тощо.

Практичні заняття мають на меті навчити студентів розв'язувати задачі з шкільного курсу основ інформатики ж за обов'язковою програмою, так і на поглибленому рівні (факультативні заняття, гуртки, класи з поглибленим вивченням інформатики, профільні класи), а також кваліфіковано використовувати засоби нової інформаційної технології для управління навчальним процесом.

Особливістю даної програми є те, що навчання студентів розв'язуванню задач органічно вплітається в методику викладання основ інформатики та обчислювальної техніки і реалізується на трьох ступенях, відповідних різним рівням інформаційної і методичної підготовленості та трьома основними ступенями становлення студента як вчителя.

Слід всіляко заохочувати студентів 2-5 курсів до участі в науково-дослідних роботах, у роботах методичних гуртків і гуртків, пов'язаних з використанням НІТ, спецкурсів і спецсеминарів, спецпрактикумів, написанні рефератів, курсових і дипломних робіт, пов'язаних з використанням засобів і методів сучасної інформаційної технології яку розв'язуванні досліджуваних проблем, так і у підготовці звітів, доповідей, опрацюванні результатів спостережень.

ПРОГРАМА КУРСУ "ОСНОВ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ" ДЛЯ ЗАГАЛЬНООСВІТНЬОЇ ШКОЛИ¹

В основу даної програми покладено діючу програму шкільного курсу "Основи інформатики та обчислювальної техніки" з урахуванням проекту державного стандарту загальної середньої освіти в Україні стосовно шкільного курсу "Основи інформатики".

Метою курсу є формування теоретичної бази знань учнів з основ інформатики та практичних навичок використання засобів сучасних інформаційних технологій у повсякденній практичній, зокрема навчально-пізнавальній, діяльності учнів.

До **теоретичної** бази знань відносяться: уявлення про інформацію, її властивості, інформаційні процеси та інформаційні системи, загальні принципи розв'язування задач за допомогою комп'ютера при використанні програм загального та прикладного забезпечення, формулювання проблем і постановка задач, побудова відповідних інформаційних (зокрема, математичних) моделей, основи алгоритмізації і програмування, принципи будови та дії комп'ютера, уявлення про можливості глобальної мережі Інтернет, пошук потрібної інформації.

До **практичних** навичок відносяться: навички роботи з пристроями введення-виведення інформації, прикладним програмним забезпеченням загального й цільового призначення: редакторами текстів, графічними редакторами, електронними таблицями, системами управління базами даних, інформаційно-пошуковими системами, педагогічними програмними засобами, програмами-броузерами для перегляду гіпертекстових сторінок; програмами для роботи з електронною поштою і телеконференціями; пошук інформації в глобальній мережі Інтернет тощо; навички складання, описування та реалізації деяких алгоритмів і програм з використанням засобів навчальної алгоритмічної мови та реальних мов програмування і операційних систем.

Мета курсу досягається через практичне оволодіння учнями навичками роботи з основними складовими сучасного програмного забезпечення ЕОМ, ознайомлення -з функціональним призначенням основних пристроїв комп'ютера та принципами їх будови і дії, основами технології розв'язування задач за допомогою комп'ютера, починаючи від їх постановки й побудови відповідних інформаційних моделей і закінчуючи інтерпретацією результатів, отриманих за допомогою комп'ютера.

Програма курсу розрахована на вивчення основ інформатики за умов достатнього доступу учнів до комп'ютерів. За

¹ Автори: Жалдак М.І., Морзе Н.В., Науменко Г.Г.

пропонованою програмою вивчення курсу сплановано на 2 роки з розрахунку 102 години. Навчальні години доцільно розподілити таким чином:

1-й рік (2 години на тиждень) - 68 годин.

2-й рік (1 година на тиждень) - 34 години.

Можливий також варіант розподілу годин, коли першого року в першому півріччі відводиться по одній годині на тиждень, а в другому - по дві; другого року - в першому півріччі відводиться по дві години на тиждень, а в другому - по одній. Залежно від варіанту навчального плану, за яким працює школа, курс "Основи інформатики і обчислювальної техніки" може вивчатись у 8-9 або в 10-11 класах.

Залежно від типу обчислювальної техніки, складу наявного науково-методичного та програмного забезпечення учитель може самостійно добирати методичні шляхи розв'язання освітніх завдань курсу, вносити необхідні корективи в порядок вивчення тем програми, а також змінювати кількість годин, необхідних для засвоєння програмного матеріалу. Окремі питання програми можуть вивчатись тільки в порядку ознайомлення. Відповідно до обраної методики викладання курсу вчитель може дібрати відповідні навчальні посібники та дидактичне забезпечення, віддаючи перевагу тим чи іншим з них або ж певним чином поєднуючи їх.

Основні складові курсу доцільно розподілити так:

Тема	Кількість
Вступ. Інформація та інформаційні процеси	3 год.
Інформаційна система	5 год.
Операційні системи	9 год.
Основи роботи з дисками	5 год.
Прикладне програмне забезпечення	46 год.
Інтернет та його основні можливості	6 год.
Основи алгоритмізації та програмування	28 год.

Для забезпечення курсу основ інформатики необхідні такі **програмні засоби**:

1. Клавіатурний тренажер - для оволодіння клавіатурою ПЕОМ.
2. Операційна система - для засвоєння правил роботи з системою введення-виведення інформації та її зберігання на зовнішніх носіях.
3. Операційна оболонка — для засвоєння правил роботи з файлами, що зберігаються на зовнішніх носіях.
4. Редактор текстів - для засвоєння правил введення, зберігання та редагування текстів на ПЕОМ.
5. Графічний редактор - для засвоєння правил виконання на ПЕОМ малюнків, схем, креслень, анімацій тощо.
6. Електронні таблиці - для засвоєння правил виконання значних за обсягом розрахунків за їх допомогою та ознайомлення з діловою графікою.
7. Системи управління базами даних - для засвоєння правил створення, редагування, впорядкування, пошуку та нагромадження інформації в базах даних.
8. Довідково-інформаційна система.
9. Набір прикладних програм цільового призначення (педагогічні програмні засоби).
10. Набір програм для роботи в глобальній мережі Інтернет - для перегляду Web-сторінок, роботи з електронною поштою та телеконференціями, файловими архівами, пошуковими машинами тощо.
11. Діалоговий інтерпретатор (компілятор) однієї з мов програмування (BASIC, PASCAL, C, Visual Basic, Delphi тощо) - для ознайомлення з правилами складання, введення, виконання та налагодження програм на ПЕОМ.

ТЕМАТИЧНЕ ПЛАНУВАННЯ НАВЧАЛЬНОГО МАТЕРІАЛУ ТА ЗМІСТ НАВЧАННЯ

Вступ. Інформація та інформаційні процеси (3 год.)

Інформатика. Поняття інформації. Інформація і повідомлення. Інформація та інформаційні процеси. Носії інформації, форми і способи подання інформації. Види інформації. Оцінка і вимірювання інформації. Інформація і шум, у взаємоперетворення. Кодування повідомлень, за допомогою яких передається інформація. Властивості інформації. Інформаційні процеси: пошук, збирання, зберігання, опрацювання, подання, передавання, використання, захист інформації. Поняття про сучасні засоби зберігання та опрацювання інформації.

Коротка історія розвитку обчислювальної техніки. Характеристика різних поколінь ЕОМ та перспективи розвитку комп'ютерної техніки. Основні компоненти ПЕОМ та їх функціональне призначення. Основні галузі застосування ЕОМ.

Учні повинні *знати*:

- поняття інформації та повідомлення, взаємозв'язки між поняттями інформація і повідомлення; види інформації; форми та засоби зберігання, подання й передавання інформації; приклади різних носіїв інформації;
- принципи та можливості кодування інформації; способи опрацювання інформації; підходи до оцінки та вимірювання інформації; поняття шуму, взаємоперетворення інформації і шуму; одиниці вимірювання ємності запам'ятовуючих пристроїв; властивості інформації; основні інформаційні процеси: пошук, збирання, зберігання, опрацювання, подання, передавання, використання, захист інформації;
- визначення інформатики як науки про засоби й методи збирання, опрацювання, зберігання, пошуку, передавання, подання та використання інформації в різних галузях людської діяльності.

Учні повинні мати *уявлення* про:

- основні галузі застосування ЕОМ;
- основні етапи історії розвитку обчислювальної техніки та перспективи розвитку комп'ютерної техніки.

Інформаційна система (5 год.)

Правила техніки безпеки при роботі в комп'ютерному класі.

Структура інформаційної системи. Апаратна та інформаційна складові інформаційної системи. Взаємодія апаратної та інформаційної складових. Основні пристрої апаратної складової інформаційної системи, їх функції та взаємодія.

Пристрої введення-виведення інформації. Клавіатура. Типи клавіатур. Дисплей. Типи, основні характеристики та принципи роботи дисплея. Принтер. Типи й принципи роботи принтерів. Сканер. Пристрої для організації комп'ютерного зв'язку. Модем, види модемів та їх функції.

Пам'ять ЕОМ. Внутрішня й зовнішня пам'ять комп'ютера. Оперативна пам'ять комп'ютера.

Процесор. Основні характеристики та функції процесора. Арифметико-логічний пристрій, пристрій управління.

Типи процесорів.

Звукові та відеоплати. Контролери. Адаптери. Магістраль.

Підготовка ПЕОМ до роботи.

Зовнішні запам'ятовуючі пристрої ПЕОМ. Дискові накопичувачі, їх основні характеристики та принципи роботи.

Розміщення повідомлень на магнітних дисках.

Основні характеристики комп'ютера. Магістрально-модульний принцип будови ПЕОМ. Принципи роботи ПЕОМ: принцип програмного управління, принцип адресності.

Огляд сучасної обчислювальної техніки. Технічні характеристики ПЕОМ.

Комп'ютерні мережі. Класифікація, основні характеристики та принципи використання комп'ютерних мереж.

Апаратні компоненти комп'ютерних мереж. Доступ до інформаційних ресурсів.

Учні повинні *знати*:

- структуру інформаційної системи;
- призначення та функції основних складових апаратної частини інформаційної системи;
- основні характеристики і функції процесора;
- види пам'яті комп'ютера;
- види пристроїв введення-виведення інформації;
- призначення контролерів, адаптерів;
- види зовнішніх запам'ятовуючих пристроїв ЕОМ;
- види дисків та їх основні характеристики;
- основні характеристики ПЕОМ;
- правила техніки безпеки при роботі в комп'ютерному класі;
- призначення основних клавіш ПЕОМ.

Учні повинні мати *уявлення* про:

- роль магістралі, принципи взаємодії основних пристроїв ЕОМ при опрацюванні інформації;
- типи процесорів;
- принципи розміщення інформації на дискових накопичувачах;
- пристрої для організації комп'ютерного зв'язку;
- принципи розміщення інформації на дискових накопичувачах;
- призначення та основні характеристики комп'ютерних мереж; типи доступу до інформаційних ресурсів.

Учні повинні *вміти*:

- готувати ПЕОМ до роботи;
- користуватися клавіатурою ПЕОМ.

Операційні системи (9 год.)

Поняття файла. Ім'я та розширення імені файла. Каталоги (папки) та підкаталоги файлів. Маршрут. Специфікація файла.

Стандартні імена зовнішніх запам'ятовуючих пристроїв ПЕОМ. Функції та склад операційної системи. Класифікація операційних систем. Повідомлення операційної системи (ОС) про готовність до роботи. Принципи роботи користувача з ОС. Інтерфейс ОС. Основні об'єкти, з якими працює ОС. Правила роботи з об'єктами. Властивості об'єктів, набір операцій над об'єктами. Запуск у роботу програм, що працюють під управлінням операційної системи. Основні вказівки операційної системи для роботи з файлами, папками, дисками. Робота з довідковою системою. Пошук файлів. Поняття про інсталяцію програмних засобів.

Налагодження роботи інформаційної системи. Конфігурація інформаційної системи.

Учні повинні *знати*:

- поняття файлу, його ім'я та розширення, каталогу (папки), підкаталогу, шляху до файлу;
- стандартні імена зовнішніх запам'ятовуючих пристроїв ПЕОМ;
- основні функції операційної системи;
- основні вказівки ОС для роботи з файлами, каталогами;
- правила запуску на виконання програми, яка працює під управлінням операційної системи;
- особливості виконуваних файлів;

Учні повинні мати *уявлення* про:

- склад операційної системи;
- конфігурування та налагодження інформаційної системи;
- правила інсталювання програмних засобів. Учні повинні вміти:

- записувати шлях до файлу;
- визначати місце знаходження потрібного файлу;
- вибирати об'єкти, з якими працює ОС; змінювати їх властивості, визначати та виконувати операції з ними;
- за допомогою вказівок операційної системи виводити на екран та на друк інформацію про файли, що знаходяться на зовнішніх носіях; копіювати і вилучати файли та папки (каталоги), перейменовувати файли та папки, виводити на екран і на друк зміст текстових файлів та документів; впорядковувати інформацію, що знаходиться в каталозі та в окремих файлах;
- відшукувати потрібні файли та в файлах потрібну інформацію;
- знаходити необхідну довідкову інформацію;
- інсталювати програмні засоби;
- запускати на виконання будь-яку програму, що працює під управлінням операційної системи.

Основи роботи з дисками (5 год.)

Форматування диска. Діагностика диска. Дефрагментація диска. Відновлення інформації на диску. Правила записування та читування інформації з дискет.

Поняття про комп'ютерні віруси. Класифікація комп'ютерних вірусів. Принципи "зараження" комп'ютерним вірусом диска й пам'яті комп'ютера. Антивірусні програми, їх класифікація та принципи роботи. Знайомство з роботою антивірусних програм. Захист інформації.

Архівація файлів. Принципи стиснення інформації. Робота з програмами-архіваторами. Створення архіву, додавання файлів до архіву, перегляд вмісту архіву, вилучення файлів із архіву, розкриття архівованих файлів.

Учні повинні *знати*:

- правила збереження інформації на диску; правила форматування дисків;; поняття про діагностику диска, правила відновлення інформації на диску; правила записування та читування інформації з дискет;
- поняття про комп'ютерний вірус; призначення антивірусних програм та їх основні можливості; правила профілактики та боротьби з комп'ютерними вірусами;
- поняття про архівований файл; правила користування та можливості програм-архіваторів. Учні повинні мати уявлення про:
- принципи форматування дисків; принцип розміщення інформації на дискових накопичувачах;
- призначення FAT-таблиці; правила запису інформації у EAT-таблиці;
- поняття про кластер та сектор;
- принципи стиснення інформації на дисках;
- принципи та шляхи захисту інформації. Учні повинні вміти:
- формувати різні диски; діагностувати диск, відновлювати інформацію на диску;
- користуватися антивірусними програмами;
- використовувати можливості програм-архіваторів, створювати архів, дописувати файли до архіву, переглядати вміст архіву, вилучати окремі файли із архіву, вибирати файли із архіву.

Прикладне програмне забезпечення загального призначення (46 год.)

Текстовий редактор (12 год.)

Системи опрацювання текстів, їх класифікація та функції. Завантаження текстового редактора. Призначення та система вказівок текстового редактора. Введення тексту з клавіатури. Редагування тексту. Перевірка орфографії. Робота з текстовими файлами. Робота з об'єктами. Використання буферу обміну. Пошук інформації в середовищі текстового редактора. Робота з контекстами - пошук, заміна, виділення, перенесення, копіювання, форматування. Робота із шрифтами. Друкування тексту. Шаблони текстів та робота з ними. Використання таблиць в текстах. Структура документа.

Графічний редактор (3 год.)

Системи опрацювання графічної інформації. Типи графічних файлів. Графічний редактор та його призначення. Робота з графічними файлами. Система вказівок графічного редактора. Вказівки малювання графічних примітивів. Створення анімацій. Вставлення малюнків до тексту.

Електронні таблиці (11 год.)

Електронні таблиці (ЕТ) та їх призначення. Введення й редагування числової, формульної та текстової інформації. Робота з файлами ЕТ. Діапазон клітинок. Опрацювання табличної інформації: копіювання, редагування, вилучення, переміщення, форматування тощо. Використання функцій та операцій для опрацювання інформації, поданої в таблиці. Ділова графіка. Побудова діаграм і графіків на основі табличної інформації. Використання статистичних функцій для опрацювання табличної інформації. Використання логічних функцій для опрацювання табличної інформації.

Створення в електронній таблиці бази даних, впорядкування та пошук потрібної інформації в середовищі ЕТ. Фільтрування даних. Аналіз даних в середовищі ЕТ.

Бази даних. Системи управління базами даних (14 год.)

Бази даних (БД). Системи управління базами даних (СУБД). Призначення та функції систем управління базою даних. Основні поняття бази даних: поле, запис, файл. Фактографічні й документальні бази даних. Ієрархічна, мережева, реляційна моделі бази даних.

Проектування БД. Створення структури БД. Типи даних, що зберігаються в БД. Введення та редагування даних. Робота з таблицями. Поняття ключового поля. Зв'язки між таблицями. Вказівки роботи з файлами в СУБД. Впорядкування даних у БД. Фільтрація даних в БД. Використання простих та складених фільтрів. Обчислювальні фільтри. Мова запитів СУБД. Пошук інформації в БД. Формування звітів. Програмування в середовищі СУБД.

Прикладні програми цільового призначення (6 год.)

Інформаційно-пошукові системи, їх призначення та робота з ними.

Прикладні програми цільового призначення. Прикладні програми для підтримки навчання іноземних мов, географії, біології, математики, фізики, комп'ютерні енциклопедії. Поняття про автоматизовані системи управління, системи автоматизованого проектування.

Учні повинні знати:

- призначення та основні функції текстового редактора; правила роботи з текстовим редактором; основні операції, які можна виконувати з текстом за допомогою текстового редактора;
- основні поняття машинної графіки; типи графічних файлів; основні операції щодо створення та редагування зображень за допомогою графічного редактора; правила роботи з графічними редакторами;
- призначення електронних таблиць; розміщення інформації в електронних таблицях; правила роботи з електронними таблицями; основні операції, які можна виконувати з даними, що містяться в електронних таблицях;
- можливості та правила опрацювання інформації, що зберігається в ЕТ;
- правила пошуку інформації та її фільтрування в середовищі ЕТ;
- визначення й призначення баз даних; основні поняття баз даних; визначення й призначення систем управління базами даних та інформаційно-пошукових систем; основні операції, які можна виконувати з даними в СУБД; правила проектування та створення БД, фільтрування та пошуку інформації в БД за допомогою спеціальної мови запитів;
- призначення інформаційно-пошукових систем;
- правила роботи з прикладними програмами спеціального призначення. Учні повинні мати уявлення про:
- типи моделей баз даних;
- принципи роботи інформаційно-пошукових систем;
- різні можливості прикладних програм спеціального призначення. Учні повинні вміти:
- завантажувати текстовий редактор, зчитувати до текстового редактора текст, який зберігається на зовнішніх носіях; з використанням текстового редактора вводити текст до запам'ятовуючих пристроїв комп'ютера, редагувати, формувати текст, зберігати текст на зовнішніх носіях, друкувати текст, відмічати блоки тексту з наступним копіюванням чи перенесенням та виконувати інші операції з контекстами, виконувати заміну одного контексту на інший, здійснювати пошук контекстів в тексті; з'єднувати кілька частин тексту до одного; підключати словник для знаходження орфографічних та граматичних помилок у тексті, вибирати й використовувати необхідний шрифт, вставляти таблиці в текст та формувати їх; використовувати шаблони документів; створювати документи за певною структурою; відшукувати текстові файли за різними ознаками: ім'ям, датою створення, автором, вмістом, обсягом тощо;
- завантажувати графічний редактор, за допомогою графічного редактора створювати малюнки, образи, динамічні фрагменти; змінювати параметри графічних об'єктів; вставляти створені малюнки до тексту;
- завантажувати програму опрацювання електронних таблиць; за допомогою програми опрацювання електронних таблиць виконувати набір операцій над даними, що зберігаються в електронній таблиці; вводити числову, формульну та текстову інформацію, використовуючи операції та функції програми опрацювання електронних таблиць, опрацьовувати табличну інформацію; будувати діаграми та графіки на основі табличної інформації;
- впорядковувати й відшукувати серед табличної інформації необхідні дані;
- фільтрувати дані, що зберігаються в ЕТ; виконувати аналіз даних, що зберігаються в ЕТ;
- завантажувати систему управління базами даних (СУБД), виконувати проектування БД; створювати структуру бази даних та заповнювати базу даних інформацією, редагувати дані у БД (вносити зміни до інформації, яка зберігається в базі даних, змінювати структуру бази даних, вилучати записи); зв'язувати данні в БД; фільтрувати та впорядковувати дані в базі даних; організовувати пошук потрібної інформації в базі даних, опрацьовувати дані різних типів за допомогою вбудованих до СУБД функцій, виконувати різні операції з файлами БД, створювати звіти, виконувати прості та складані запити в БД;
- запускати на виконання та використовувати для розв'язування задач з відповідних предметних галузей програми для підтримки розв'язування задач з математики, фізики, хімії, навчання мов, біології, географії, історії та інших дисциплін.

Глобальна мережа Інтернет та її можливості (6 год.)

Глобальна мережа Інтернет. Різні типи під'єднання комп'ютерів до глобальної мережі. Сервер та робоча станція. Адресація в Інтернет. Основні сервіси глобальної мережі Інтернет: WWW, електронна пошта, телеконференції, файлові архіви. Ідентифікація комп'ютерів в мережі. Протоколи передавання інформації. Принципи функціонування глобальної мережі. Провайдери та інформація, необхідна для під'єднання до мережі Internet. Види та основні можливості програмного забезпечення для роботи в глобальній мережі Інтернет.

Електронна пошта. Принципи функціонування. Поштові стандарти. Електронна адреса. Основні можливості поштових програм для роботи з електронними повідомленнями: створення електронного повідомлення, відправлення, відправлення копій, приєднання файлів до повідомлень, одержання повідомлення тощо. Адресна книга.

Поняття про телеконференції. Програмне забезпечення та конфігурація телеконференцій. Різні групи телеконференцій. Правила організації і проведення телеконференції. Пошук потрібних груп новин та підписки на них. Відправлення повідомлень в телеконференції.

Служба World Wide Web (WWW). Організація інформації, принципи навігації. Поняття про формат документів гіпертекстових документів. WWW-сервери. Адреса Web-сторінки. Пошук інформації в Інтернет. Програми-броузери. Правила роботи та налагодження. Використання різноманітних сторінок кодування. Файлові ресурси в мережі Інтернет.

Проблеми безпеки та захисту інформації в глобальній мережі Інтернет.

Учні повинні знати:

- можливості основних сервісів глобальної мережі; види програмного забезпечення, необхідного для роботи в

- глобальній мережі Інтернет;
 - поняття про гіпертекст та правила роботи з ним; принципи адресації в WWW; принципи та правила здійснення пошуку інформації в Інтернет;
 - типи та можливості програм для підтримки роботи електронної пошти;
 - правила електронного листування; правила роботи з адресною книгою;
 - формати, за допомогою яких можна відправляти графічну та звукову інформацію;
 - особливості роботи користувачів з телеконференціями; правила спілкування за допомогою телеконференцій;
- Учні повинні мати *уявлення* про:

- основні принципи будови і функціонування сучасних глобальних комп'ютерних мереж; типи під'єднання комп'ютерів до глобальної мережі;
- поняття комунікаційного протоколу;
- принципи функціонування електронної пошти; правила утворення електронної адреси; правила використання різноманітних сторінок кодування;
- правила організації і роботи з телеконференціями; основні групи телеконференцій. Учні повинні *вміти*:
- запускати на виконання програму-броузер; переглядати гіпертекстові сторінки; працювати з програмами броузерами, створювати закладки на потрібних Web-сторінках; переміщуватися по сторінках в прямому і зворотному напрямках; вводити з клавіатури адресу потрібної Web-сторінки; здійснювати пошук потрібної інформації в Інтернет;
- запускати на виконання програму роботи з електронною поштою, яка працює під управлінням операційної системи DOS і Windows; складати, редагувати і відправляти через комп'ютерну мережу електронні листи;
- давати відповідь на електронні листи; одержувати пошту, користуватися адресною книгою, приєднувати до електронних повідомлень файли; перекодовувати повідомлення, одержані через електронну пошту; виконувати переадресацію поштових повідомлень;
- підключатися до електронних конференцій, розміщувати там власну інформацію і читати необхідну інформацію, переписувати інформацію, що є в різних конференціях; підписуватися на потрібну телеконференцію; спілкуватися з іншими учасниками телеконференцій;
- здійснювати пошук потрібних файлових архівів; копіювати із файл-сервера файлові архіви; копіювати на файл-сервер файлові архіви.

Основи алгоритмізації та програмування (28 год.)

Інформаційна модель (1 год.)

Основні етапи розв'язування прикладної задачі з використанням ЕОМ. Поняття інформаційної (математичної) моделі. Побудова неформальної моделі: визначення положень, на яких буде ґрунтуватися неформальна модель, визначення вхідних даних і результатів, запис співвідношень, які пов'язують вхідні дані й результати.

Алгоритми (5 год.)

Поняття про алгоритм. Приклади алгоритмів. Виконавці алгоритмів. Властивості базових структур алгоритмів. Способи описання алгоритмів.

Структурний підхід до побудови алгоритмів. Технологія структурного програмування. Метод покрокової деталізації. Розробка алгоритму зверху донизу. Модульна побудова алгоритму. Поняття про аналіз алгоритмів. Поняття про метод послідовного уточнення при побудові алгоритму.

Навчальна алгоритмічна мова (НАМ). Алфавіт. Синтаксис. Основні правила описання алгоритмів. Основні вказівки НАМ. Величини. Основні характеристики величин. Поняття про алгоритми роботи з величинами. Аргументи та результати алгоритму. Присвоювання значення величині.

Програма. Мова програмування (7 год.)

Поняття програми. Поняття про мови програмування. Класифікація мов програмування. Поняття про системи програмування. Поняття про інтерпретацію та компіляцію. Інтегровані середовища програмування. Поняття редактора, транслятора, налагоджувача.

Мова програмування. Особливості та призначення. Переклад програм з однієї мови на іншу. Алфавіт. Основні поняття мови: оператори, ідентифікатори, числа, рядки, описи. Величини. Типи даних. Стандартні типи даних. Набір функцій і операцій, введених для кожного з стандартних типів. Вирази. Пріоритет операцій. Оператори. Оператор присвоювання. Оператори введення та виведення. Вказівка присвоювання. Опис програми мовою програмування.

Звернення до алгоритмів і функцій (1 год.)

Опис навчальною алгоритмічною мовою та мовою програмування вказівок звернення до алгоритмів і функцій. Формальні та фактичні параметри.

Вказівки повторення й розгалуження (6 год.)

Опис вказівок повторення й розгалуження навчальною алгоритмічною мовою та мовою програмування. Опис умов.

Табличні величини (4 год.)

Табличні величини. Алгоритми роботи з табличними величинами. Методи впорядкування та пошуку.

Рядкові величини (4 год.)

Рядкові величини. Алгоритми роботи з рядками. Опис найпростіших алгоритмів роботи з рядками мовою програмування.

Учні повинні *знати*:

- основні етапи розв'язування задачі з використанням ЕОМ; поняття інформаційної моделі задачі;
- поняття алгоритму, властивості алгоритму, способи та форми подання алгоритму; основні базові структури алгоритмів; сутність методу послідовного уточнення алгоритму;

- основні ідеї та принципи технології структурного програмування;
- порядок складання алгоритмів і програм;
- правила запису структурованих алгоритмів і програм;
- основні вказівки навчальної алгоритмічної мови;
- основні елементи однієї з мов програмування:
- алфавіт; основні поняття мови: числа, рядки, описи, ідентифікатори, оператори, величини, операції;
- типи даних у мові програмування, набір функцій і операцій, допустимих для кожного з типів даних;
- принципи побудови опису програми (мовою Pascal чи Basic);
- сутність операції присвоювання; призначення та правила описування вказівок розгалуження й повторення; звернення до алгоритмів; поняття про алгоритми-процедури та алгоритми-функції. особливості використання табличних та рядкових величин.

Учні повинні мати *уявлення* про:

- класифікацію мов програмування; поняття про інтерпретацію та компіляцію; систему програмування;
- інтегровані середовища програмування;

Учні повинні *вміти*:

- застосовувати різні форми опису алгоритмів і переходити від однієї форми опису алгоритмів до іншої;
- визначати тип величини, описувати її навчальною алгоритмічною мовою та мовою програмування; використовувати прості й складні умови при побудові алгоритмів і програм; застосовувати метод послідовних уточнень при побудові алгоритмів і програм;
- описувати алгоритми розв'язування задач різних типів навчальною алгоритмічною мовою та мовою програмування;
- складати й реалізовувати найпростіші лінійні, розгалужені, циклічні алгоритми: на опрацювання табличних і рядкових величин, на опрацювання графічної інформації; на використання допоміжних функцій і процедур;
- налагоджувати програми на ПЕОМ.

ОРІЄНТОВНЕ ПОУРОЧНЕ ПЛАНУВАННЯ КУРСУ "ОСНОВИ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ" 1 варіант (для комп'ютерів, які працюють під управлінням ОС Windows)

1-й РІК (68 год.)

1. Вступ. Інформація та інформаційні процеси (3 год.)
 - 1.1. Інформатика. Поняття інформації. Інформація і повідомлення. Інформація та інформаційні процеси. Носії інформації, форми і способи подання інформації. Види інформації. Оцінка і вимірювання інформації. Інформація і шум та їх взаємоперетворення.
 - 1.2. Кодування повідомлень, за допомогою яких передається інформація. Властивості інформації. Інформаційні процеси: пошук, збирання, зберігання, передавання, опрацювання, використання, захист інформації. Поняття про сучасні засоби зберігання та опрацювання інформації.
 - 1.3. Коротка історія розвитку обчислювальної техніки. Характеристика різних поколінь ЕОМ. Основні компоненти ЕОМ та їх функціональне призначення. Основні галузі застосування ЕОМ.
2. Інформаційна система (5 год.)
 - 2.1. Техніка безпеки при роботі на ЕОМ. Структура інформаційної системи. Апаратна та програмна складові інформаційної системи. Основні пристрої апаратної складової інформаційної системи, їх функції та взаємодія. Взаємодія апаратної та програмної складових. Процесор. Основні функції процесора.
 - 2.2. Пристрої введення-виведення інформації. Клавіатура. Типи клавіатур. Дисплей. Типи, основні характеристики та принципи роботи дисплею. Принтер. Сканер. Пристрої для організації комп'ютерного зв'язку. Модем. Правила підготовки ПЕОМ до роботи. Клавіатурний тренажер.
 - 2.3. Практична робота №1. "Робота з клавіатурним тренажером".
 - 2.4. Пам'ять ЕОМ. Внутрішня пам'ять комп'ютера. Зовнішні запам'ятовуючі пристрої ПЕОМ. Дискові накопичувачі, їх основні характеристики та принципи роботи. Розміщення інформації на магнітних дисках.
 - 2.5. Основні характеристики комп'ютера. Магістрально-модульний принцип будови ПЕОМ. Принципи роботи ЕОМ: принцип програмного управління, принцип адресності. Огляд сучасної обчислювальної техніки. Технічні характеристики ПЕОМ. Комп'ютерні мережі. Класифікація, основні характеристики та принципи використання комп'ютерних мереж. Апаратні компоненти комп'ютерних мереж. Доступ до інформаційних ресурсів.
3. Операційні системи (9 год.)
 - 3.1. Поняття файла. Ім'я та розширення імені файла. Каталоги (папки) файлів. Шлях до файла. Специфікація файлів. Стандартні імена зовнішніх запам'ятовуючих пристроїв ПЕОМ. Функції та склад операційної системи. Класифікація операційних систем персонального комп'ютера. Операційна система Windows 98. Завантаження та завершення роботи з ОС Windows 98. Інтерфейс ОС.
 - 3.2. Робочий стіл. Основні об'єкти, з якими працює ОС. Виділення об'єкта. Операції, властивості та основні команди для роботи з об'єктами ОС. Контекстне меню об'єкта. Ярлики та їх призначення.
 - 3.3. Принципи роботи з вікнами. Основні типи вікон: прикладні, підлеглі, діалогові, інформаційні. Основні елементи вікон, їх призначення та правила роботи з ними. Меню.
 - 3.4. Практична робота №2. "Робота з об'єктами на робочому столі: виділення, переміщення, перегляд властивостей, створення, перейменування, вилучення. Робота з вікнами."
 - 3.5. Основне меню ОС Windows 98 - кнопка Пуск. Правила виклику довідкової інформації. Пошук інформації в

- середовищі ОС Windows 98. Запуск програм, що працюють під управлінням ОС Windows 98. Закінчення роботи з програмою. Запуск на виконання музичних записів. Використання калькулятора, блокнота, графічного редактора.
- 3.6. Практична робота №3. "Виклик довідкової інформації. Пошук потрібної інформації. Запуск на виконання програм, що працюють під управлінням ОС Windows 98."
- 3.7. Програма Провідник. Виконання операцій з файлами та папками: виділення, виклик властивостей, перегляд, створення, копіювання, перенесення, вилучення, перейменування, друкування, запуск на виконання тощо. Вказівки операційної системи для роботи з файлами. Поняття про інсталювання програмних засобів.
- 3.8. Практична робота №4. "Вказівки операційної системи з файлами та папками. Інсталювання програм."
- 3.9. Комплексна практична робота №5. "Операційна система Windows 98."
4. Основи роботи з дисками (5 год.)
- 4.1. Форматування диска. Діагностика диску. Дефрагментація диска. Відновлення інформації на диску. Правила записування та зчитування інформації з дискет. Поняття про комп'ютерні віруси. Класифікація вірусів. Антивірусні програми.
- 4.2. Практична робота №6. "Робота з дисками. Форматування, сканування, дефрагментація, запис інформації. Робота з антивірусними програмами."
- 4.3. Архівація файлів. Принципи стиснення інформації. Програми для роботи з архівами. Створення архівних файлів. Додавання інформації до архіву. Перегляд вмісту архіву. Розкриття файлів із архіву.
- 4.4. Практична робота №7. "Архівація файлів".
- 4.5. Комплексна практична робота №8. "Робота з дисками".
5. Прикладне програмне забезпечення загального призначення (46 год.)
- 5.1. Системи опрацювання текстів і їх основні функції. Завантаження текстового редактора (ТР). Інтерфейс ТР. Інформаційний рядок. Режими екрана, використання вікон. Одержання довідок. Переміщення в тексті. Завантаження тексту з зовнішніх носіїв до ОЗП.
- 5.2. Основні об'єкти при роботі з текстовим редактором. Контекстне меню. Режим вставлення та заміни. Перемикання клавіатур. Редагування тексту. Перевірка орфографічних та граматичних помилок. Пошук інформації в середовищі ТР.
- 5.3. Практична робота №9. "Редагування тексту".
- 5.4. Робота з основним меню текстового редактора. Збереження тексту на зовнішніх носіях. Пошук документа на диску. Створення нового документа.
- 5.5. Практична робота №10. "Введення тексту з клавіатури".
- 5.6. Форматування символів тексту: вибір шрифту, зміна розмірів літер, накреслення, кольору, встановлення ефектів. Форматування абзаців: зміна відступів, типу вирівнювання. Робота зі сторінками.
- 5.7. Практична робота №11. "Форматування тексту".
- 5.8. Робота з фрагментами тексту: виділення, вставлення, переміщення, копіювання, вилучення, пошук і заміна фрагментів тексту. Використання буферу обміну.
- 5.9. Вставляння таблиць до текстового документу. Форматування таблиць.
- 5.10. Практична робота №12. "Робота з таблицями в середовищі текстового редактора".
- 5.11. Шаблони документів. Структура документа. Друкування тексту.
- 5.12. Комплексна практична робота №13. "Робота в середовищі текстового редактора".
- 5.13. Машинна графіка. Графічний екран. Система опрацювання графічної інформації. Типи графічних файлів. Вказівки створення графічних примітивів при роботі з графічним редактором.
- 5.14. Створення малюнків та анімацій за допомогою графічного редактора. Вставляння малюнків до тексту.
- 5.15. Практична робота №14. "Робота з графічним редактором".
- 5.16. Програми опрацювання електронних таблиць. Призначення і функції електронних таблиць. Введення й редагування інформації в електронній таблиці. Форматування таблиць.
- 5.17. Практична робота №15. "Введення, редагування та форматування табличної інформації".
- 5.18. Введення формул до електронної таблиці. Опрацювання табличної інформації.
- 5.19. Практична робота №16. "Опрацювання табличної інформації за допомогою вбудованих функцій та операцій табличного процесора".
- 5.20. Використання логічних формул і операцій при опрацюванні табличної інформації.
- 5.21. Практична робота №17. "Опрацювання табличної інформації за допомогою логічних формул і операцій".
- 5.22. Ділова графіка. Побудова діаграм і графіків при роботі з табличним процесором.
- 5.23. Практична робота №18. "Побудова діаграм і графіків при роботі з табличним процесором".
- 5.24. Підтримка баз даних в табличному процесорі. Впорядкування даних. Фільтр простий та розширений. Пошук інформації.
- 5.25. Практична робота №19. "Створення бази даних в табличному процесорі. Впорядкування та пошук даних в табличному процесорі".
- 5.26. Комплексна практична робота №20. "Опрацювання та пошук табличної інформації в середовищі ЕТ".
- 5.27. Бази даних. Фактографічні й документальні бази даних. Ієрархічна, мережева, реляційна модель бази даних. Основні елементи бази даних: поле, запис, файл. Системи управління базами даних. Знайомство й робота з базою даних. Пошук інформації в базі даних.
- 5.28. Проектування бази даних. Створення структури реляційної бази даних. Типи даних, які використовуються в БД. Введення даних.
- 5.29. Практична робота №21. "Створення структури бази даних і введення даних".

- 5.30. Редагування даних. Редагування структури бази даних. Робота з таблицями в різних режимах: режимі конструктора таблиць та режимі роботи з таблицями.
- 5.31. Впорядкування даних, пошук даних за зразком в таблиці. Використання простих фільтрів для пошуку даних в БД.
- 5.32. Практична робота №22. "Впорядкування даних та пошук даних в БД".
- 5.33. Типи зв'язків в таблицях. Ключові поля. Створення зв'язків між елементами в таблицях.
- 5.34. Форми. Призначення та режими роботи з формами. Створення форм.
- 5.35. Звіти. Призначення та режими роботи із звітами. Створення звітів.
- 5.36. Практична робота №23. "Створення форм та звітів."
- 5.37. Запити. Призначення. Різниця запитів і фільтрів. Створення простих запитів.
- 5.38. Практична робота №24. "Створення простих запитів".
- 5.39. Комплексна практична робота №25. "Робота в середовищі СУБД".
- 5.40. Програмні засоби цільового призначення. Інформаційно-пошукові системи. Знайомство й робота з інформаційно-пошуковою системою.
- 5.41. Програмні засоби для підтримки навчання географії.
- 5.42. Програмні засоби для підтримки навчання іноземної мови.
- 5.43. Програмні засоби для підтримки навчання біології.
- 5.44. Програмні засоби для підтримки навчання математики.
- 5.45. Залік.

2-рік (34 год.)

6. Глобальна мережа Інтернет та її можливості (6 год.)
 - 6.1. Глобальна мережа Інтернет. Різні типи під'єднання комп'ютерів до глобальної мережі. Сервер та робоча станція. Адресація в Інтернет. Протоколи передавання інформації, які підтримують роботу в глобальній мережі. Принципи функціонування глобальної мережі. Провайдери та інформація, необхідна для під'єднання до мережі Інтернет. Основні сервіси глобальної мережі Інтернет: WWW, електронна пошта, телеконференції, файлові архіви. Види та основні можливості програмного забезпечення для роботи в глобальній мережі Інтернет.
 - 6.2. Електронна пошта в глобальній мережі. Принципи функціонування. Поштові стандарти. Електронна адреса. Основні можливості поштових програм для роботи з електронними повідомленнями: створення електронного повідомлення, відправлення, відправлення копій, приєднання файлів до повідомлень, одержання повідомлення тощо. Адресна книга.
 - 6.3. Практична робота №26. "Створення, відправлення та одержання електронного повідомлення".
 - 6.4. Поняття про телеконференції. Програмне забезпечення та конфігурація телеконференцій. Різні групи телеконференцій. Правила організації і проведення телеконференцій. Пошук потрібних груп новин та підписки на них. Відправлення повідомлень в телеконференції. Практична робота №27. "Робота з телеконференціями".
 - 6.5. Служба World Wide Web (WWW). Організація інформації, принципи навігації. Поняття про формат гіпертекстових документів. WWW-сервери. Адреса Web-сторінки. Програми-броузери. Правила роботи та налагодження. Використання різноманітних сторінок кодування. Файлові ресурси в Інтернет.
 - 6.6. Пошук інформації в Інтернет. Пошукові системи. Практична робота №28. "Пошук інформації в Інтернет".
7. Основи алгоритмізації та програмування (28 год.)
 - 7.1. Основні етапи розв'язування прикладної задачі з використанням ЕОМ. Поняття інформаційної моделі. Побудова моделі.
 - 7.2. Алгоритми. Властивості алгоритмів. Форми подання алгоритму. Виконавець алгоритму.
 - 7.3. Базові структури алгоритмів. Метод покрокової деталізації. Конструювання алгоритмів "зверху донизу".
 - 7.4. Навчальна алгоритмічна мова. Правила описання алгоритмів навчальною алгоритмічною мовою.
 - 7.5. Величини. Основні характеристики величин. Вказівка присвоювання.
 - 7.6. Поняття програми. Мова програмування. Переклад програм із однієї мови на іншу. Поняття про транслятори, систему програмування.
 - 7.7. Поняття про середовище програмування. Запуск програм на виконання.
 - 7.8. Практична робота №29. "Робота в середовищі програмування. Запуск програм на виконання".
 - 7.9. Алфавіт мови програмування. Величини та їх опис мовою програмування. Змінні та константи. Ідентифікатори величин. Типи величин. Вказівки введення й виведення. Вказівка присвоювання.
 - 7.10. Стандартні типи змінних. Набір функцій та операцій, що визначені для кожного із стандартних типів. Арифметичні вирази. Пріоритет операцій.
 - 7.11. Опис програм за правилами мови програмування. Переклад лінійних алгоритмів з навчальної алгоритмічної мови на мову програмування.
 - 7.12. Практична робота №30. "Створення і реалізація найпростіших лінійних програм".
 - 7.13. Вказівки виклику алгоритму. Створення алгоритмів із використанням звернень до алгоритмів.
 - 7.14. Практична робота №31. "Складання та реалізація програм, що містять звернення до алгоритмів".
 - 7.15. Вказівка повторення та її опис навчальною алгоритмічною мовою і мовою програмування.
 - 7.16. Опис циклічних алгоритмів навчальною алгоритмічною мовою. Переклад циклічних алгоритмів із навчальної алгоритмічної мови на мову програмування.
 - 7.17. Практична робота №32. "Опис і реалізація найпростіших циклічних програм".
 - 7.18. Вказівка розгалуження та її опис навчальною алгоритмічною мовою і мовою програмування.

- 7.19. Опис алгоритмів із розгалуженням навчальною алгоритмічною мовою та мовою програмування.
- 7.20. Практична робота №33. "Створення та реалізація програм із розгалуженнями".
- 7.21. Табличні величини та їх опис навчальною алгоритмічною мовою і мовою програмування. Алгоритми знаходження суми й добутку елементів таблиць.
- 7.22. Алгоритм пошуку в таблицях елементів із деякою властивістю. Алгоритми впорядкування табличних величин.
- 7.23. Практична робота №34. "Створення та реалізація програм опрацювання табличних величин".
- 7.24. Вказівки і функції опрацювання рядкових величин та їх опис навчальною алгоритмічною мовою і мовою програмування.
- 7.25. Практична робота №35. "Складання та реалізація програм опрацювання рядкових величин".
- 7.26. Комплексна практична робота №36.
- 7.27. Залік.

ОРІЄНТОВНЕ ПОУРОЧНЕ ПЛАНУВАННЯ КУРСУ "ОСНОВИ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ" 2 варіант (для комп'ютерів, які працюють під управлінням операційної системи MS-DOS)

1-й РІК (68 год.)

1. Вступ. Інформація та інформаційні процеси (3 год.)
 - 1.1. Інформатика. Поняття інформації. Інформація і повідомлення. Інформація та інформаційні процеси. Носії інформації, форми і способи подання інформації. Види інформації. Оцінка і вимірювання інформації. Інформація і шум та їх взаємоперетворення.
 - 1.2. Кодування повідомлень, за допомогою яких передається інформація. Властивості інформації. Інформаційні процеси: пошук, збирання, збереження, передавання, опрацювання, використання, захист інформації. Поняття про сучасні засоби зберігання та опрацювання інформації.
 - 1.3. Коротка історія розвитку обчислювальної техніки. Характеристика різних поколінь ЕОМ. Основні компоненти ЕОМ та їх функціональне призначення. Основні галузі застосування ЕОМ.
2. Інформаційна система (5 год.)
 - 2.1. Техніка безпеки при роботі на ЕОМ. Структура інформаційної системи. Апаратна та програмна складові інформаційної системи. Основні пристрої апаратної складової інформаційної системи, їх функції та взаємодія. Взаємодія апаратної та програмної складових. Процесор. Основні функції процесора.
 - 2.2. Пристрої введення-виведення інформації. Клавіатура. Типи клавіатур. Дисплей. Типи, основні характеристики та принципи роботи дисплею. Принтер. Сканер. Пристрої для організації комп'ютерного зв'язку. Правила підготовки ПЕОМ до роботи. Клавіатурний тренажер.
 - 2.3. Практична робота №1. "Робота з клавіатурним тренажером".
 - 2.4. Пам'ять ЕОМ. Внутрішня пам'ять комп'ютера. Зовнішні запам'ятовуючі пристрої ПЕОМ. Дискові накопичувачі, їх основні характеристики та принципи роботи. Розміщення інформації на магнітних дисках.
 - 2.5. Основні характеристики комп'ютера. Магістрально-модульний принцип будови ПЕОМ. Принципи роботи ЕОМ: принцип програмного управління, принцип адресності. Огляд сучасної обчислювальної техніки. Технічні характеристики ПЕОМ. Комп'ютерні мережі. Класифікація, основні характеристики та принципи використання комп'ютерних мереж. Апаратні компоненти комп'ютерних мереж. Доступ до інформаційних ресурсів.
3. Операційні системи (11 год.)
 - 3.1. Поняття файлу. Ім'я та розширення імені файлу. Каталоги й підкаталоги файлів. Шлях до файлу. Специфікація файлів. Стандартні імена зовнішніх запам'ятовуючих пристроїв ПЕОМ.
 - 3.2. Функції та склад операційної системи. Класифікація операційних систем. Операційні системи персональних комп'ютерів. Модулі ОС, їх призначення. Завантаження операційної системи. Системний диск.
 - 3.3. Вказівки ОС для роботи з файлами.
 - 3.4. Сервісні вказівки ОС. Інсталювання програм.
 - 3.5. Практична робота №2. "Вказівки операційної системи. Інсталювання програм."
 - 3.6. Командні файли.
 - 3.7. Операційна оболонка. Завантаження операційної оболонки. Вказівки та функції операційної оболонки.
 - 3.8. Вказівки операційної оболонки для роботи з файлами.
 - 3.9. Практична робота №3. "Вказівки операційної оболонки для роботи з файлами".
 - 3.10. Додаткові вказівки операційної оболонки. Зміна конфігурації та використання додаткових параметрів операційної оболонки.
 - 3.11. Практична робота №4. "Додаткові вказівки операційної оболонки. Використання додаткових параметрів операційної оболонки".
4. Основи роботи з дисками (5 год.)
 - 4.1. Форматування диска. Діагностика диска. Дефрагментація диска. Правила записування та зчитування інформації з дискет. Відновлення інформації на диску.
 - 4.2. Поняття про комп'ютерні віруси. Класифікація вірусів. Антивірусні програми.
 - 4.3. Архівація файлів. Принципи стиснення інформації. Програми для роботи з архівами. Створення архівних файлів. Додавання інформації до архіву. Перегляд архіву. Розкриття файлів із архіву.
 - 4.4. Практична робота №5. "Архівація файлів".
 - 4.5. Комплексна практична робота №6. "Робота з дисками".
5. Прикладне програмне забезпечення загального призначення (44 год.)

- 5.1. Системи опрацювання текстів і їх основні функції. Завантаження текстового редактора. Інтерфейс текстового редактора. Інформаційний рядок текстового редактора. Режими екрана, використання вікон. Одержання довідок. Переміщення в тексті. Завантаження тексту з зовнішніх носіїв до ОЗП.
- 5.2. Режим вставлення та заміни. Перемикання клавіатур. Редагування тексту. Перевірка орфографічних та граматичних помилок.
- 5.3. Практична робота №7. "Редагування тексту",
- 5.4. Робота з основним меню текстового редактора. Збереження тексту на зовнішніх носіях. Пошук документа на диску. Створення нового документа.
- 5.5. Практична робота №8. "Введення тексту з клавіатури".
- 5.6. Форматування символів тексту: вибір шрифту, зміна розмірів букв, накреслення, кольору, встановлення ефектів. Форматування абзаців: зміна відступів, типу вирівнювання. Робота зі сторінками.
- 5.7. Практична робота №9. "Форматування тексту".
- 5.8. Робота з фрагментами тексту: виділення, вставлення, переміщення, копіювання, видалення, пошук і заміна фрагментів тексту. Використання буферу обміну. Друкування тексту.
- 5.9. Практична робота №10. "Робота з фрагментами тексту".
- 5.10. Вставлення таблиць до текстового документа. Форматування таблиць.
- 5.11. Комплексна практична робота №11. "Робота в середовищі текстового редактора".
- 5.12. Машинна графіка. Графічний екран. Система опрацювання графічної інформації. Типи графічних файлів. Вказівки малювання графічних примітивів при роботі з графічним редактором.
- 5.13. Створення малюнків та анімацій за допомогою графічного редактора.
- 5.14. Практична робота №12. "Робота з графічним редактором".
- 5.15. Програми опрацювання електронних таблиць. Призначення і функції електронних таблиць. Введення й редагування інформації в електронній таблиці. Форматування таблиць.
- 5.16. Практична робота №13. "Введення, редагування та форматування табличної інформації".
- 5.17. Введення формул до електронної таблиці. Опрацювання табличної інформації.
- 5.18. Практична робота №14. "Опрацювання табличної інформації за допомогою вбудованих функцій та операцій табличного процесора".
- 5.19. Використання логічних формул і операцій при опрацюванні табличної інформації.
- 5.20. Практична робота №15. "Опрацювання табличної інформації за допомогою логічних формул і операцій".
- 5.21. Ділова графіка. Побудова діаграм і графіків при роботі з табличним процесором.
- 5.22. Практична робота №16. "Побудова діаграм і графіків при роботі з табличним процесором".
- 5.23. Підтримка баз даних в табличному процесорі. Впорядкування даних. Фільтр простий та розширений. Пошук інформації.
- 5.24. Практична робота №17. "Створення бази даних в табличному процесорі. Пошук даних в табличному процесорі".
- 5.25. Комплексна практична робота №18. "Опрацювання та пошук табличної інформації в середовищі ЕТ".
- 5.26. Бази даних. Фактографічні й документальні бази даних. Ієрархічна, мережева, реляційна модель бази даних. Основні елементи бази даних: поле, запис, файл. Системи управління базами даних. Знайомство й робота з базою даних. Пошук у базі даних.
- 5.27. Створення структури бази даних. Введення даних.
- 5.28. Практична робота №19. "Створення структури бази даних і введення даних".
- 5.29. Формат вказівки для роботи з базою даних. Діапазон дії вказівки. Вказівки виведення інформації на екран. Запис умов у вказівках СУБД.
- 5.30. Редагування даних. Редагування структури бази даних.
- 5.31. Практична робота №20. "Редагування даних і структури бази даних".
- 5.32. Вказівки роботи з файлами бази даних.
- 5.33. Практична робота №21. "Робота з файлами бази даних".
- 5.34. Впорядкування даних. Пошук інформації в базі даних.
- 5.35. Практична робота №22. "Впорядкування та пошук інформації в базі даних".
- 5.36. Вказівки введення й виведення інформації. Створення командних файлів у СУБД.
- 5.37. Практична робота №23. "Створення найпростіших командних файлів".
- 5.38. Програмні засоби цільового призначення. Інформаційно-пошукові системи. Знайомство й робота з інформаційно-пошуковою системою.
- 5.39. Прикладні програми з географії.
- 5.40. Прикладні програми з іноземної мови.
- 5.41. Прикладні програми з біології.
- 5.42. Прикладні програми з математики.
- 5.43. Комплексна практична робота №24. "Робота з програмними засобами цільового призначення".

2-рік (34 год.)

6. Глобальна мережа Інтернет та її можливості. (5 год.)
 - 6.1. Глобальна мережа Інтернет. Різні типи під'єднання комп'ютерів до глобальної мережі. Сервер та робоча станція. Адресація в Інтернет. Протоколи передавання інформації, які підтримують роботу в глобальній мережі. Принципи функціонування глобальної мережі. Провайдери та інформація, необхідна для під'єднання до мережі Інтернет. Основні сервіси глобальної мережі Інтернет: WWW, електронна пошта, телеконференції, файлові архіви. Види та

- основні можливості програмного забезпечення для роботи в глобальній мережі Інтернет.
- 6.2. Електронна пошта в глобальній мережі. Принципи функціонування. Поштові стандарти. Електронна адреса. Основні можливості поштових програм для роботи з електронними повідомленнями: створення електронного повідомлення, відправлення, відправлення копій, приєднання файлів до повідомлень, одержання повідомлення тощо. Адресна книга.
 - 6.3. Практична робота №25. "Створення, відправлення та одержання електронного повідомлення".
 - 6.4. Поняття про телеконференції. Програмне забезпечення та конфігурація телеконференцій. Різні групи телеконференцій. Правила організації і проведення телеконференцій. Пошук потрібних груп новин та підписки на них. Відправлення повідомлень в телеконференції.
 - 6.5. Служба World Wide Web (WWW). Організація інформації, принципи навігації. Поняття про формат гіпертекстових документів. WWW-сервери. Адреса Web-сторінки. Програми-броузери. Правила роботи та налагодження. Використання різноманітних сторінок кодування. Файлові ресурси в Інтернет. Пошук інформації в Інтернет. Пошуківі системи.
7. Основи алгоритмізації та програмування (29 год.)
- 7.1. Основні етапи розв'язування прикладної задачі з використанням ЕОМ. Поняття інформаційної моделі. Побудова моделі.
 - 7.2. Алгоритми. Властивості алгоритмів. Форми подання алгоритму. Виконавець алгоритму.
 - 7.3. Базові структури алгоритмів.
 - 7.4. Навчальна алгоритмічна мова. Правила описання алгоритмів навчальною алгоритмічною мовою.
 - 7.5. Величини. Основні характеристики величин. Вказівка присвоювання.
 - 7.6. Поняття програми. Мова програмування. Переклад програм із однієї мови на іншу. Поняття про транслятори, систему програмування.
 - 7.7. Поняття про середовище програмування. Запуск програм на виконання.
 - 7.8. Практична робота №26. "Робота в середовищі програмування. Запуск програм на виконання".
 - 7.9. Алфавіт мови програмування. Величини та їх опис мовою програмування. Змінні та константи. Ідентифікатори величин. Типи величин. Вказівки введення й виведення. Вказівка присвоювання.
 - 7.10. Стандартні типи змінних. Набір функцій та операцій, що визначені для кожного із стандартних типів. Арифметичні вирази. Пріоритет операцій.
 - 7.11. Опис програм за правилами мови програмування. Переклад лінійних алгоритмів з навчальної алгоритмічної мови на мову програмування.
 - 7.12. Практична робота №27. "Створення і реалізація найпростіших лінійних програм".
 - 7.13. Вказівки виклику алгоритму. Створення алгоритмів із використанням звернень до алгоритмів.
 - 7.14. Практична робота №28. "Складання та реалізація програм, що містять звернення до алгоритмів".
 - 7.15. Вказівка повторення та її опис навчальною алгоритмічною мовою і мовою програмування.
 - 7.16. Опис циклічних алгоритмів навчальною алгоритмічною мовою. Переклад циклічних алгоритмів із навчальної алгоритмічної мови на мову програмування.
 - 7.17. Практична робота №29. "Опис і реалізація найпростіших циклічних програм".
 - 7.18. Вказівка розгалуження та її опис навчальною алгоритмічною мовою і мовою програмування.
 - 7.19. Опис алгоритмів із розгалуженням навчальною алгоритмічною мовою та мовою програмування.
 - 7.20. Практична робота №30. "Створення та реалізація програм із розгалуженнями".
 - 7.21. Табличні величини та їх опис навчальною алгоритмічною мовою і мовою програмування. Алгоритми знаходження суми й добутку елементів таблиць.
 - 7.22. Алгоритм пошуку в таблицях елементів із деякою властивістю. Алгоритми впорядкування табличних величин.
 - 7.23. Створення та реалізація програм опрацювання табличних величин.
 - 7.24. Практична робота №31. "Створення та реалізація програм опрацювання табличних величин".
 - 7.25. Вказівки і функції опрацювання рядкових величин та їх опис навчальною алгоритмічною мовою і мовою програмування.
 - 7.26. Практична робота №32. "Складання та реалізація програм опрацювання рядкових величин".
 - 7.27. Залік.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Бауэр Ф.Л., Гооз Г. Информатика. - М.: Мир, 1990. -Т. 1, 2.
2. Есаян А.Р. и др. Информатика: Учеб. пособие для педагогических институтов. - М.: Просвещение, 1991. - 288 с.
3. Жалдак М.І., Рамський Ю.С. Информатика. – К.: Вища школа, 1991.
4. Заварыкин В.М., Житомирский В.Г., Лапчик М.П. Основы информатики и вычислительной техники. – М.: Просвещение, 1989.
5. Зарецька І.Т. та ін. Информатика: Навч. посіб. – К., 2002. – 496 с.
6. Кеймин В.А. и др. Основы информатики и вычислительной техники. - М.: Просвещение, 1989. - 275 с.
7. Міхеев В.В. Методика навчання інформатики: Методичних посібник для студ. вищих пед. навч. закл. – Житомир: Поліграфічний центр ЖДПУ, 2004. – 224 с.: іл.
8. Міхеев В.В. Основы информатики: Методичних посібник для студ. вищих навч. закл. – Житомир: Поліграфічний центр ЖДПУ, 2005. – 124 с.: іл.
9. Ситник В.Ф. та ін.. Основы информационных систем: Навчальний посібник. — К.: КНЕУ, 2001. — 420 с.
10. Спірін О.М. Практична інформатика. – Житомир, 1999.

11. Чалий О.В., Дяков В.А., Хаїмзон І.І. Основи інформатики. – К.: Вища школа, 1993.
12. Шкиль Н.И., Жалдак М.И., Морзе Н.В., Рамский Ю.С. Изучение языков программирования в школе. - К.: Рад. шк., 1989.

**ПОЛОЖЕННЯ
ПРО РЕЙТИНГОВУ СИСТЕМУ ОЦІНКИ ЗНАНЬ, ВМІНЬ ТА НАВИЧОК СТУДЕНТІВ
ПРИ ВИВЧЕННІ КУРСУ “МЕТОДИКА НАВЧАННЯ ІНФОРМАТИКИ”**

1. Загальна можлива набрана сума балів – 100, з яких 50 балів складає екзамен.
2. Оцінка “задовільно” ставиться, якщо студент набрав в сумі від 61 до 74 балів.
Оцінка “добре” ставиться, якщо студент набрав в сумі від 75 до 90 балів.
Оцінка “відмінно” ставиться, якщо студент набрав в сумі від 91 до 100 балів.
Студент допускається до складання заліку, якщо він має за роботу під час 1-го семестру не менше 20 балів.
Студент допускається до складання екзамену, якщо він має за роботу під час навчання не менше 31 балів.
3. Присутність студента на занятті під час виконання лабораторної роботи оцінюється в 1 бал (максимальна кількість 19 балів). Якщо робота розрахована на декілька занять, то 1 бал поділяється пропорційно.
4. Виконана та вчасно захищена лабораторна робота оцінюється в 1 бал (загальна кількість 19 балів). Невчасно захищена лабораторна робота оцінюється в 0,5 бали.
5. Якщо при захисті роботи студент виявив добрі та відмінні теоретичні знання або результати самостійної роботи та зразково і творчо виконав практичну складову роботи, то робота може бути додатково оцінена в 0,5 – 1,5 бали.
6. Контрольна робота оцінюється в 1 – 12 балів.
7. Якщо загальна сума балів за результатами навчання перевищує 61, то студент за його бажанням та згодою викладача може отримати відповідну підсумкову оцінку без складання екзамену.

ЛАБОРАТОРНІ РОБОТИ

ЛАБОРАТОРНА РОБОТА №1 (4 год.)

Календарно-тематичний план вчителя інформатики

ОСНОВНА МЕТА	Ознайомлення з календарно-тематичним планом вчителя інформатики. Формування вмінь та навичок по складанню календарно-тематичного плану.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ	Розуміти роль і структуру календарно-тематичного плану. Вміти складати план роботи вчителя інформатики.

1. Теоретичні відомості та методичні рекомендації

Мета та зміст шкільного курсу інформатики визначаються *Програмою з інформатики для загальноосвітньої школи*. При цьому передбачено три варіанти вивчення інформатики в загальноосвітній школі.

1) Повний курс (140 год., машинний варіант з використанням комп'ютерів, що працюють під управлінням ОС WINDOWS).

2) Повний курс (140 год., машинний варіант з використанням комп'ютерів, що працюють під управлінням ОС MS-DOS).

Ці варіанти призначені для шкіл, що забезпечені сучасною комп'ютерною технікою чи мають можливість організувати систематичні заняття у комп'ютерному класі іншої організації.

3) Скорочений курс (70 год., безмашинний варіант).

При роботі за машинним варіантом клас ділиться на дві групи, але не менше 8 учнів у групі (наказ Міністерства освіти і науки України від 20.02.2002 року №128).

За рішенням місцевих органів виконавчої влади або органів місцевого самоврядування класи можуть ділитися на групи і при наповнюваності меншій від нормативної, за рахунок бюджетних асигнувань та залучення додаткових коштів.

У разі розбіжності кількості годин у робочому навчальному плані навчального закладу з діючою програмою, вчителям інформатики дозволяється, за погодженням з відповідними районними (міськими) методичними кабінетами, вносити корективи до програми з інформатики, відповідно до навчального плану закладу. *При цьому вчитель повинен забезпечувати виконання навчальної програми у повному обсязі.*

Курс інформатики може вивчатись в 7—9 класах за авторськими програмами, що мають гриф Міністерства освіти і науки України, якщо в робочому навчальному плані закладу передбачено вивчення предмету «Інформатика». В іншому випадку для опанування комп'ютерною грамотністю в 7—9 класах можуть бути введені спецкурси. В 1—6 класах може вводитись пропедевтичний курс інформатики за такими напрямками:

- *загальноосвітній*, в якому інформатика розглядається як засіб розвитку логічного мислення, формування вмінь аналізувати, описувати плани дій та робити логічні висновки (для цього немає необхідності використовувати комп'ютери);
- *технологічний*, тобто навчання конкретним інформаційним технологіям: вироблення основних навичок користування комп'ютером, використання комп'ютерів як інструментів для загальноосвітніх цілей (випуск шкільних газет та журналів, малювання, комп'ютерне листування, пошук інформації тощо).

Спецкурси та пропедевтичні курси вивчаються за рахунок годин варіативної частини навчального плану закладу за програмами, складеними вчителем та затвердженими директором навчального закладу.

У спеціалізованих школах (класах), гімназіях, ліцеях, колегіумах, зокрема з поглибленим вивченням курсу інформатики, години трудового навчання (технології) та навчально-виробничої практики можуть використовуватися для вивчення спецкурсів професійного спрямування за діючими програмами або за програмами, погодженими з відповідними районними (міськими) методичними кабінетами.

Навчання учнів інформатиці має здійснюватися систематично, послідовно, з урахуванням вимог програми з інформатики, згідно з кількістю годин, визначених навчальним планом школи. Плани роботи вчителя інформатики можуть відбивати навчально-виховний процес, систему позакласної роботи та роботу кабінету.

Для організації навчально-виховного процесу вчитель повинен уміти складати плани: перспективний (річний), тематичний, план-конспект уроку.

Перспективний (річний) план відтворює вивчення інформатики за темами. Він дозволяє визначити місце теми в структурі предмету, спланувати практичні заняття, систему повторення, провести підготовчу роботу до теми, враховуючи певні засоби навчання. Перспективний план потрібний кожному вчителю-початківцю для самоконтролю в строках проходження матеріалу, використання обов'язкових форм навчання, організації систематичного внутрішньопредметного і міжпредметного повторення. Він може складатися за формою 1.

Назва теми	Позаурочна робота	Кількість годин	Термін проведення	Повторення матеріалу		Лабораторна робота	Позакласне заняття
				Внутрішньо предметне	Міжпредметне		
1	2	3	4	5	6	7	8

Тематичний план дозволяє бачити місце кожного уроку в системі уроків теми, визначати значення кожного кроку в процесі формування умінь на основі знань, відображає систему вивчення школярами прийомів і методів самостійної навчальної діяльності, дає можливість вчителю своєчасно підготувати необхідні засоби навчання до кожного уроку.

Основні **вимоги** до календарно-тематичного планування:

- 1) тематичний план складається на основі програми і в основному відповідає їй;
- 2) матеріал по уроках повинен бути розподілений рівномірно і послідовно, в логічному взаємозв'язку;
- 3) матеріал по можливості повинен відповідати підручнику або основному посібнику;
- 4) таблиця для складання календарно-тематичного плану може мати такі колонки (форма 2):

№ з/п	Тема уроку	Кіль-ть год.	Дата	Обладнання та програмне забезпечення	Примітка
1	2	3	4	5	6

Тематичний план дає змогу вчителю побачити всю тему вцілому, визначити послідовність формування понять, практичних умінь, методів і прийомів самостійної роботи учнів. Він допомагає своєчасно організувати підготовку, проведення програмних практичних занять, які мають попередній характер, а також систематично здійснювати внутрішньо-предметні і міжпредметні зв'язки в процесі оволодіння школярами знаннями з інформатики.

Тематичний план можна скласти і за такою схемою (див. приклад *Програми з інформатики для загальноосвітньої школи*):

1. Назва теми і кількість годин, які даються шкільною програмою на її вивчення.
2. Система понять теми.
3. Навчально-виховні завдання теми:
 - освітні;
 - розвиваючі;
 - виховні.

Розподіл уроків теми здійснюється за такою формою 3:

№ п/п	Тема уроку	Тип уроку	Вид уроку	Лабораторні роботи	Обладнання та комп'ютерне програмне забезпечення	Самостійна робота учнів	Питання для повторення та узагальнення	Література	
								Для вчителя	Для учнів
1	2	3	4	5	6	7	8	9	10

Під час складання тематичного плану особливу увагу потрібно звернути на визначення завдань теми. Плануючи освітні завдання, слід виділити систему понять, які треба заново сформувати, і понять, які розвиваються в поданій темі. Окремо треба відзначити розвиток загальних понять, а також необхідно виділити формування або розвиток спеціальних і загальнонавчальних умінь і навичок.

Плануючи розвиваючі завдання теми, необхідно передбачити розвиток мислення, пам'яті, уваги учнів, формування мотивів і потреби в навчанні, розвиток пізнавальних інтересів і здібностей учнів під час вивчення конкретної теми.

Під час планування виховних завдань на конкретну тему треба вибирати тільки ті з них, реалізація яких можлива з урахуванням особливостей змісту теми, методів і прийомів, які використовуються.

Такий комплекс взаємопов'язаних завдань теми дозволяє отримати кращі навчально-виховні результати.

Заповнюючи форму, слід користуватись програмою з шкільного курсу інформатики і підручником.

У графі 7 слід відбити можливі види позаурочної самостійної роботи учнів у зв'язку з вивченням матеріалу уроку: роботу з підручником, з комп'ютером, з натуральними і образотворчими засобами наочності, робочим зошитом.

У графу 8 потрібно включити той конкретний матеріал програми з інформатики, який необхідно повторити у зв'язку з вивченням нових понять (внутрішньопредметні зв'язки), і той матеріал з інших предметів, який допоможе засвоїти ці поняття (міжпредметні зв'язки).

У графі 9 зазначається література з теми для вчителя, а в графі 10 – література, рекомендована учням для позаурочного та позакласного читання.

2. Завдання до лабораторної роботи

1. Ознайомитися з шкільною програмою з інформатики (див. *Програму курсу інформатики для загальноосвітньої школи*).

2. Скласти календарно-тематичний план на 1-й семестр 10 класу, враховуючи орієнтовне поурочне планування курсу інформатики для класів, що мають комп'ютери, які працюють під управлінням ОС Windows та шкільний підручник або навчальний посібник за формою 2.

3. Скласти тематичний план вивчення теми “Операційні системи” 1-го семестру 10 класу, враховуючи орієнтовне поурочне планування курсу інформатики для класів, що мають комп'ютери, які працюють під управлінням ОС MS-DOS, шкільний підручник або навчальний посібник та наявні педагогічні програмні засоби пакету “Методика навчання інформатики” (див. лаб. роб. №3) за формою 3.

Примітка:

- залежно від типу обчислювальної техніки, складу наявного науково-методичного та програмного забезпечення учитель може самостійно добирати методичні шляхи розв'язання освітніх завдань курсу, вносити необхідні корективи в порядок вивчення тем програми, а також змінювати кількість годин, необхідних для засвоєння програмного матеріалу. Окремі питання програми можуть вивчатися тільки в порядку ознайомлення. Відповідно до обраної методики викладання курсу вчитель може дібрати відповідні навчальні посібники та дидактичне забезпечення, віддаючи перевагу тим чи іншим з них або ж певним чином поєднуючи їх;
- при плануванні враховуйте наявність самостійних та контрольних робіт, уроків розв'язування задач, заліків тощо.

4. Оформити звіт по лабораторній роботі.

ЛАБОРАТОРНА РОБОТА №2 (4 год.)

Конспект уроку вчителя інформатики

ОСНОВНА МЕТА	Формування вмінь та навичок по складанню конспекту уроку.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ	Розуміти роль і структуру конспекту уроку. Вміти складати розширений конспект уроку з інформатики.

1. Теоретичні відомості та методичні рекомендації

Урок – основна форма організації навчально-виховної роботи в школі.

Основні **характеристики** уроку:

- мета;
- зміст;
- методи і засоби навчання;
- організація навчальної діяльності.

Мета – загальноосвітня, виховна, практична (розвиваюча).

Методи навчання – це упорядковані способи взаємопов'язаної діяльності вчителя і учнів, які направлені на досягнення цілей навчання.

Опис кожного метода **вимагає**:

- 1) опис навчальної діяльності вчителя;
- 2) опис навчальної (пізнавальної) діяльності учня;
- 3) опис зв'язку між ними, або спосіб, яким навчаюча діяльність вчителя керує пізнавальною діяльністю учня.

При виборі певного методу слід **враховувати**:

- 1) мету уроку;
- 2) структуру та інші особливості змісту матеріалу уроку (складність, новизна, характер матеріалу тощо);
- 3) особливості учнів класу (рівень розвитку мислення; рівень знань, вмінь та навичок; рівень вихованості тощо);
- 4) наявність певних дидактичних матеріалів, обладнання, комп'ютерного програмного забезпечення і т. ін.;
- 5) час проведення уроку, наповненість класу тощо;
- 6) індивідуальну майстерність вчителя та його особисті людські якості.

Методи навчання поділять на *методи наукового дослідження* та *методи навчання* конкретного предмету (зокрема, *інформатики*).

Методи наукового дослідження складаються з загальних методів дидактики з врахуванням специфіки навчального предмету (індукція, дедукція, аналіз, синтез, аналогія, узагальнення, абстрагування, конкретизація, класифікація, систематизація) і спеціальних (притаманних тільки методиці навчання інформатики) методів (метод побудови математичних та інформаційних моделей, специфічні методи абстрагування тощо).

До методів навчання інформатики належать проблемний метод, проблемне навчання через задачі, евристичний метод, алгоритмічний метод, програмове навчання, конкретно-індуктивний та абстрактно-дедуктивний методи тощо.

Застосування **проблемного методу** передбачає:

- 1) створення проблемної ситуації;
- 2) певна готовність та певний інтерес у учнів до відшукування розв'язку;
- 3) можливість неоднозначного шляху розв'язку проблеми.

Схема реалізації проблемного методу:

- 1) створення проблемної ситуації;
- 2) висунення гіпотез учнями;
- 3) обговорення гіпотез;
- 4) розв'язування проблеми;
- 5) висновки.

На уроках з інформатики слід використовувати загальні, групові та індивідуальні **форми** роботи з учнями.

Форми організації навчальної діяльності: лекції, семінари, практичні заняття, консультації, конференції, диспути, дослідна робота, додаткові заняття з відстаючими учнями тощо.

Типи уроків. Уроки можна класифікувати за двома ознаками – за *дидактичною метою* та за *способом їх проведення*.

За дидактичною метою:

- урок вивчення нового матеріалу;
- урок закріплення вивченого;
- урок перевірки знань, вмінь та навичок;
- урок систематизації та узагальнення вивченого матеріалу.

За способом проведення:

- комбінований урок;
- урок пояснення нового матеріалу;
- урок розв'язування задач;

- оглядовий урок;
- урок повторення;
- урок-лекція;
- урок-семінар;
- лабораторна робота;
- урок-залік;
- самостійна робота;
- контрольна робота;
- урок-бесіда тощо.

Вимоги до уроку інформатики:

- ціленаправленість;
- раціональна побудова змісту уроку;
- оптимальний вибір засобів, методів та прийомів навчання і виховання.

Основні етапи уроку:

- постановка мети уроку перед учнями;
- ознайомлення з новим матеріалом;
- закріплення вивченого;
- перевірка знань, вмінь та навичок учнів;
- систематизація та узагальнення вивченого матеріалу.

Основні типи уроків.

Структура комбінованого уроку.

- Перевірка домашнього завдання;
- пояснення нового матеріалу;
- закріплення вивченого матеріалу;
- інструкція до домашнього завдання.

Види уроків вивчення нового матеріалу.

- Пояснення нового матеріалу вчителем;
- самостійна робота учнів з підручниками, посібниками, іншою додатковою літературою;
- самостійна робота учнів з програмним забезпеченням;
- самостійне вивчення матеріалу в ході розв'язування задач тощо.

Види уроків лабораторної роботи.

- Розробка алгоритмів розв'язку задач, програм і т. ін.;
- робота з комп'ютерним програмним забезпеченням;
- обчислювальна обробка результатів задач, алгоритмів тощо;
- робота з таблицями, довідковою літературою, включаючи підручники і спеціальні *інструкції*, що складені за такою *схемою*.

- 1) *Тема роботи.*
- 2) *Мета роботи.*
- 3) *Обладнання та програмне забезпечення.*
- 4) *Короткі теоретичні відомості.*
- 5) *Завдання до лабораторної роботи та план роботи.*
- 6) *Форма представлення результатів.*
- 7) *Контрольні запитання.*
- 8) *Вимоги до оцінювання результатів роботи.*

Лекційно-практична система.

Лекція в школі відрізняється від лекції у вищому навчальному закладі. Шкільна лекція повинна бути близькою до бесіди, де можна заслуговувати короткі повідомлення та коментарі учнів. В лекції треба поєднувати інформаційні та проблемні аспекти. Лекція в школі не повинна підмінювати роботу учнів з підручником та іншою літературою.

Оптимально розглядати шкільну лекцію як елемент лекційно-практичної системи, що має таку *структуру*:

- підготовчий урок;
- лекція;
- практичні заняття;
- семінари;
- контрольні-залікові уроки.

Семінар – це колективне обговорення теми під керівництвом вчителя.

Семінар, в основному, проводиться для систематизації, узагальнення і закріплення знань учнів. Основний навчально-виховний ефект має підготовча робота учнів до семінару: опрацювання матеріал за підручником і додатковою літературою; розв'язування задач; самостійне складання алгоритмів, програм, вправ, задач; виготовлення наочності, комп'ютерного програмного забезпечення; підготовка повідомлень, виступів, рефератів.

Для ефективної організації роботи семінар повинен відповідати таким *вимогам*:

- 1) тема семінару повинна бути ключовою;
- 2) тема повинна викликати інтерес і бути посилюююю для самостійного опрацювання;
- 3) наявність відповідної літератури та іншого методичного та програмного забезпечення;
- 4) необхідний запас знань, вмінь та навичок учня.

Конспект уроку

План уроку визначає систему понять, що вивчаються, на кожний урок, формування загальнонавчальних і спеціальних умінь відповідно до змісту, відображає вирішення комплексу виховних завдань, розвиток особистісних якостей учнів.

Учитель розкриває у плані всі структурні елементи уроку, ретельно продумує кожний етап, визначає методи і засоби подання матеріалу.

Методичні **вимоги** до уроку інформатики.

1. Сформулювати тему уроку.
2. Визначити цілі уроку: освітні, практичні (розвиваючі) і виховні.
3. Визначити тип уроку і відповідно до нього розробити структуру уроку, визначити конкретну задачу кожного етапу уроку.
4. Відібрати навчальний матеріал відповідно до мети (освітньої) і задачами окремих етапів уроку.
5. Визначити методи і прийоми роботи учнів.
6. Визначити прийоми керівництва вчителем діяльністю учнів.
7. Вибрати засоби навчання (підручник, дидактичні або роздавальні матеріали й ін.).
8. Визначити форму і зміст матеріалів для перевірки засвоєння матеріалу, розглянутого на уроці.
9. Продумати інструктаж до виконання домашнього завдання і форму підведення підсумків уроку.

У **розгорнутому плані** уроку повинні знайти відображення наступні моменти: тема уроку, цілі уроку, обладнання та програмне забезпечення уроку, етапи уроку (у плані повинні бути позначені тільки ті з них, що будуть на конкретному уроці).

Методичні рекомендації та вимоги до різних етапів уроку:

1. *Перевірка домашнього завдання.* У плані вказати тільки те, що перевіряється, з якою метою й у якій формі.
2. *Пояснення (вивчення) нового матеріалу.* У плані вказати:
 - а) головну суть нового матеріалу (про що матеріал, який вид задач, який прийом (алгоритм, спосіб, метод) вводиться і т.п.);
 - б) яким дидактичним прийомом (способом, методом) буде вивчатися нове: читання книги, пояснення вчителя, доповіді учнів, самостійне розв'язування задач, робота з комп'ютером і т.п.;
 - в) якщо подається новий алгоритм або вводиться новий прийом розв'язування задач, то відповідний матеріал повинен бути викладений в плані.
3. *Організація самостійної роботи учнів.* У плані указати форми організації самостійної роботи (дано завдання працювати з книгою і сформульовані питання, на які треба в книзі знайти відповідь; дано завдання розв'язати задачі і приведені вказівки на комп'ютері, що допомагає й організує розв'язування задач; дані два варіанти задач і зазначені аналогічні, але уже розв'язані задачі, якими можуть скористатися учні у випадку утруднення, і т.п.; наведено розв'язок задач).
4. *Перевірка самостійної роботи учнів.* У плані повинна бути зазначена одна з форм перевірки: учитель збере виконану роботу і перевірить після уроку; учні обмінюються зошитами і виконують взаємоперевірку; на комп'ютері буде пред'явлене правильний розв'язок і учні звіряться зі зразком і т.п.
5. *Домашнє завдання.* У плані повинна бути зазначена мета домашнього завдання, визначена форма його пред'явлення учнями і розв'язані всі задачі, що даються їм додому.
6. *Висновки по уроку.* У цій частині плану повинне бути чітко сформульовано, які пізнавальні результати одержать учні на уроці (довідаються новий факт, познайомляться з операційним складом нової навчальної дії, навчаться якихось елементів оцінки або самооцінки і т.п.). Висновки по уроку тісно пов'язані з цілями уроку, але вони уточнюють і конкретизують мету і, природно, одержують закінчене формулювання тільки в реально проведеному уроці. Якщо поставлені цілі до уроку були досягнуті, то, звичайно, висновки будуть їх у своїй основі повторювати, тільки конкретизуючи. Якщо ж у ході уроку довелося відхилитися від основної мети, то і висновки будуть інші.

2. Завдання до лабораторної роботи

1. Розробити розширений конспект уроку з вказаної нижче теми, враховуючи методичні вимоги до конспекту уроку з інформатики та наступні рекомендації і тезові допоміжні матеріали.

Тема уроку: "Комп'ютерні віруси та боротьба з ними. Антивірусні програми."

Попередня тема: "Архівація даних" (див. методичні рекомендації та теоретичні відомості до лаб. роб. №8).

Тип уроку: комбінований.

Обладнання та ТЗН: комп'ютерна техніка, таблиці.

Програмне забезпечення: антивірусна програма Dr. Web.

Теоретичні відомості:

Історичні і економічні причини появи вірусів.

Інформація є основою світової економіки. Інформація - це гроші. Лише 10% часу гроші існують у своєму фізичному вигляді, весь інший час вони існують у вигляді інформації.

Оскільки, інформаційні системи все ширше використовуються для фінансових операцій на всіх рівнях, то природно чекати, що на всіх рівнях певні елементи і групи будуть атакувати їх з метою отримання доходів незаконним шляхом. Дуже розвинуте промислове шпигунство, як міждержавне, так і серед конкуруючих фірм. Найнебезпечнішою може бути інформаційна війна, в ході якої можна проводити стратегічні наступальні операції по руйнуванню інформаційного простору держави, які в свою чергу можуть призвести до повного руйнування систем керування державою, економікою і військами.

За даними асоціації ICSA (International Computer Security Association) на кінець 1997 року, імовірність зараження корпоративних настільних систем, портативних комп'ютерів і мереж за рік збільшилася в три рази, а за два роки – в 20 разів. Практично, будь-яка організація піддається вірусній атаці хоч би один раз на рік.

Кількість вірусів збільшується щорічно. В 1990 році було відомо 500 вірусів, зараз – сотні тисяч.

Дійсним стихійним лихом стали макровіруси. Вони заражають документи і шаблони, що створені додатками Microsoft Office.

Сьогодні на долю макровірусів приходить до 80% всіх заражень вірусом в корпораціях. За даними ICSA кожен напад вірусу обходиться в 8366 доларів. Ще дорожчий час, що витрачений на відновлення системи – в середньому це 44 години і 21,7 людино-днів роботи.

Комп'ютерні віруси.

Комп'ютерні віруси - це особлива категорія невеликих за розміром програм, що можуть приписувати себе до інших програм, здатні самостійно поширюватися в комп'ютерному середовищі та виконувати небажані дії на комп'ютері.

Програма, всередині якої знаходиться вірус, називається інфікованою.

Коли заражена вірусом програма викликається для виконання, на одному із етапів її роботи вірус перехоплює керування і реалізує функції, що закладені в неї розробником. Традиційно таких функцій дві: інфікування інших програм і дисків та враження інфікованих систем.

Стадію активних дій вірусу по враженню інфікованих систем називають атакою вірусу.

В зв'язку із зараженням комп'ютерним вірусом перед користувачем ПК стоять три проблеми:

- як виявити інфікування;
- як виявити вірус;
- як позбавитися від вірусу.

Методи профілактики від вражень вірусом.

Універсальних рецептів для профілактики від вражень вірусів не існує, але деякі рекомендації можна дати:

1. Утримуйтеся від випадкових контактів і зв'язків, що виражаються у використанні чужих комп'ютерів і дискет, стерильність яких можна ставити під сумнів.
2. Зберігайте програми і дані на різних дискетах або в різних підкаталогах жорсткого диску. Вставляючи свої дискети в чужий ПК, закривайте вікна, щоб не було можливості запису на дискету.
3. Обмежуйте доступ до файлів програм, встановлюючи, коли це можливо, статус "лише для читання".
4. При роботі в мережі по можливості не викликайте програми із пам'яті інших комп'ютерів.
5. Постійно користуйтеся програмами перевірки наявності вірусів.

Програмні засоби для боротьби із вірусами.

В світі не існує єдиної класифікації вірусів на сьогоднішній день. Однак існують спроби класифікувати всі відомі на сьогодні віруси в певні групи за певними ознаками. Відповідно до цих класифікацій і відділяють певні групи антивірусних засобів.

Можна виділити три групи засобів боротьби із вірусами:

- *сканери* – антивірусні програми, що найбільш повно і надійно визначають присутність вірусу, а потім лікують інфіковані об'єкти видаляючи з них тіло вірусу і відновлюючи початковий стан об'єкта. Ці програми мають в своїх базах даних десятки тисяч масок вірусів і потужний механізм евристичного аналізу, який дозволяє їм знаходити невідомі віруси, що з'явилися пізніше, ніж противірусні програми;

- *ревізори* – антивірусні програми, які не мають в своїй базі масок вірусів; вони їх просто не знають. Але в своїй базі даних вони зберігають найбільш повну інформацію про файли, що зберігаються на даному комп'ютері або в локальній мережі. Їх завдання – виявити вірус, а лікуванням займаються програми-сканери;

- *монітори* – антивірусні програми, що знаходяться в пам'яті і контролюють всі процеси, що відбуваються на комп'ютері. Ці програми не дуже популярні, бо постійно знаходяться в пам'яті комп'ютера і сповільняють його роботу. Поряд з програмами-ревізорами існують апаратно-програмні засоби, що забезпечують більш надійний захист від проникнення вірусу. Вони блокують будь-які спроби вірусів внести зміни в системні області і контрольовані файли.

Ні одна програма не гарантує повного захисту від вірусів, тому потрібно противірусні програми використовувати в комплексі. Серед найбільш популярних на вітчизняному комп'ютерному ринку слід виділити сканери-ревізори Dr.Web, антивірус Касперського, Norton Antivirus, ревізор Adinf.

2. Провести ділову гру "Урок інформатики в школі" за вказаною темою зі студентами групи.

3. Провести **аналіз** поданого уроку за **схемою**:

- 1) визначити ключові моменти уроку: місце даного уроку в системі уроків по темі; мету і задачі уроку; тип і структуру уроку.
- 2) провести аналіз кожної структурної частини уроку:
 - визначити, яка освітня задача вирішується;
 - проаналізувати зміст відібраного навчального матеріалу й установити відповідність його поставленій задачі;
 - відзначити методи і прийоми, використані вчителем (розповідь, бесіда, створення проблемної ситуації, постановка проблемних питань і т.д.);
 - форми організації діяльності учнів (колективна й індивідуальна робота, усна і письмова робота);
 - форми і методи контролю за діяльністю учнів.
- 3) виконати аналіз дидактичних і психологічних особливостей уроку за схемою комплексного аналізу уроку. Для цього відповісти на наступні питання.
 - Чи правильно визначена мета уроку?
 - Чи враховані освітні та розвиваючі задачі, що розв'язувались на уроці?
 - Чи обґрунтований добір навчального матеріалу на урок? (Відповідність змісту уроку його основній навчальній меті, чи достатній обсяг навчального матеріалу на уроці, чи відбитий в матеріалі взаємозв'язок теорії і практики?).
 - Які методи і прийоми роботи використовував учитель на уроці, чи відповідають вони меті і змістові навчального

матеріалу? (Постановка проблемних питань, їхня точність і логічність).

- Розв'язування пізнавальних задач, їхня доступність; формування вміння працювати з книгою, комп'ютером і т.д.)
- Які методи і прийоми контролю й оцінки діяльності учнів використовував учитель на уроці?
- Які особливості діяльності учнів на уроці? (Стійкість уваги учнів, швидкість переключення уваги; зовнішні прояви активної розумової діяльності; співвідношення роботи за зразком і самостійної роботи учнів; розумові дії, якими володіють учні; позитивні і негативні емоції учнів на уроці; відношення окремих учнів до навчальної роботи на уроці.)

4) зробити загальні висновки по уроку:

- організація уроку;
- досягнення мети і задач уроку (освітніх, розвиваючих і виховних);
- загальна оцінка діяльності вчителя й учнів на уроці.

4. Оформити звіт по лабораторній роботі.

ЛАБОРАТОРНА РОБОТА №3 (4 год.)

Програмне забезпечення ПЕОМ при навчанні інформатики

ОСНОВНА МЕТА	Знайомство з основним програмним забезпеченням шкільного курсу інформатики.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ	Знати призначення та структуру програмного забезпечення шкільного курсу інформатики.

1. Теоретичні відомості та методичні рекомендації

Кабінет інформатики повинен містити:

- базовий комплект обчислювальної техніки, який включає 4-15 робочих місць;
- локальна мережа;
- загальнодоступний запам'ятовуючий пристрій з ємністю не менше 500 Кб;
- друкуючий пристрій;
- базовий комплект програмного забезпечення;
- базовий комплект документації;
- інша додаткова та методична література;
- засоби наочності;
- аптечка;
- базовий комплект протипожежної безпеки;
- інструкція з техніки безпеки;
- журнал з техніки безпеки;
- демонстраційна робоча дошка, що не потребує крейди.

Вимоги до обчислювальної техніки в освіті:

- висока універсальність та адаптованість;
- наявність в складі системи ряду робочих місць;
- забезпечення одночасної незалежної роботи на робочих місцях;
- забезпечення взаємозв'язку для організації спільної роботи;
- забезпечення можливості протоколювання ходу розв'язку задачі на кожному робочому місці;
- наявність розвинутих засобів відображення графічної і текстової інформації;
- простота і доступність взаємодії учнів з ПЕОМ;
- адаптованість до вікових та індивідуальних особливостей;
- простота і надійність при експлуатації не підготовленими користувачами;
- захищеність від випадкових і навмисних неправильних дій учнів;
- гігієнічність і безпека;
- мінімальна вартість;
- простота і зручність у користуванні.

Програмне забезпечення – це сукупність програм із усією інформацією, що відноситься до них, і яка дозволяє використовувати ЕОМ для розв'язку різноманітного типу задач.

Програмне забезпечення ЕОМ можна поділити на системне, прикладне, кросове та тестове.



Системне програмне забезпечення – це сукупність програмних і мовних засобів, які забезпечують функціонування самої обчислювальної системи, реалізують сервісні функції комп'ютера по відношенню до користувача та створюють зручне середовище для розробки інших програм.

По функціональному призначенню в системному програмному забезпеченні виділяють операційну систему, систему програмування та сервісні програми.

Операційна система – це комплекс програм, що забезпечують керування апаратурою ЕОМ, планування ефективного використання ресурсів та розв'язок задач користувача.

Система програмування – це сукупність програмних засобів, що забезпечують автоматизацію розробки і налагодження програм.

Програмні компоненти системи програмування функціонують під керуванням ОС так само, як прикладні програми користувача.

До *сервісних програм* відносять оболонки, утиліти, антивірусні програми та інші програми, що розширюють можливості системи.

Кросове програмне забезпечення – це програмне забезпечення, що призначене для роботи з певною ЕОМ і її програмами, але реалізоване на ЕОМ інших класів або типів. Його використовують, наприклад, при проектуванні й створенні нових ЕОМ.

Тестове програмне забезпечення – це сукупність програм діагностики та обслуговування апаратного забезпечення комп'ютера, що призначені для перевірки працездатності пристроїв, що входять в склад ЕОМ, як у процесі виготовлення, так і при експлуатації.

Прикладне програмне забезпечення – це сукупність програм, що призначені для розв'язку конкретних задач користувача.

Прикладні програми дозволяють фахівцям з будь-якої галузі використовувати комп'ютери у своїй професійній роботі без досконалого вивчення інформатики.

У складі ППЗ *загального користування* виділяють: текстові та графічні редактори, електронні таблиці, системи керування базами даних, системи підтримки комунікацій.

Текстові редактори призначені для автоматизації процесу набору текстових документів, їх редагування, форматування, верстки, збереження на зовнішніх носіях, роздрукування. Сучасні текстові редактори дозволяють формувати таблиці, вставляти в тексти графічні та інші нетекстові об'єкти, містять засоби перевірки орфографії та граматики тощо. Найуживанішими текстовими редакторами є Lexicon (для ОС MS DOS), Microsoft Word (для ОС Windows) тощо.

Графічні редактори служать для побудови графічних зображень, їх редагування, перегляду, збереження у вигляді файлів, роздрукування. Сучасні графічні редактори дозволяють здійснювати анімацію зображень, створити різноманітні просторові ефекти. Відомими графічними редакторами є Paint Brush (для ОС MS DOS), Corel Draw (для ОС Windows) тощо.

Електронні таблиці – це програми для табличного подання та опрацювання таких видів даних, як числа, тексти та формули. Популярними електронними таблицями є Super Calc (для ОС MS DOS) та Microsoft Excel (для Windows).

Системи керування базами даних (СУБД) призначені для створення баз даних та роботи з ними. Найуживанішими СУБД є Карат, FoxPro (для ОС MS DOS), Microsoft Access (для ОС Windows) тощо.

Системи підтримки комунікацій призначені для підключення до комп'ютера різних пристроїв, приладів та установок, а також для організації комп'ютерного зв'язку. Зв'язок між комп'ютерами найчастіше функціонує у вигляді мереж: локальних та глобальних.

До *проблемно-орієнтовного* програмного забезпечення відносять програми, які використовують у сферах практичної діяльності, і орієнтовні на використання спеціалізованих методів подання та обробки даних, властивих для певної області. Сюди відносять математичні пакети програм, системи психологічного тестування, автоматизовані навчальні системи, системи автоматизованого проектування та автоматизовані системи управління (АСУ) технологічними процесами тощо.

Інтегровані пакети – це пакети програм, компоненти яких є програмами загального користування. Інтегровані пакети мають єдине уніфіковане середовище, у межах якого можна працювати з даними, що подані різними способами. При цьому всі програми із складу пакету використовують один і той же принцип кодування даних для їх запису на диски (формат даних). Це дає можливість фрагменти даних переносити з однієї програми в іншу. Відмітимо, що в ОС MS DOS для перенесення даних між програмами, які розроблені різними фірмами, потрібно було використовувати спеціальні програми перекодування даних, якщо відповідно імпортно-експортних можливостей не мали самі прикладні програми. У межах одного інтегрованого пакету ця проблема знімається.

Зручним у використанні для MS DOS є інтегрований пакет Framework.

ОС Windows містить буфер обміну, який можна використовувати для перенесення даних між програмами. Разом з тим для ОС Windows існує інтегрований пакет Microsoft Office, який включає такі спеціалізовані програми: Microsoft Word, Microsoft Excel, Microsoft Access, Microsoft PowerPoint, Outlook Express та інші.

Вимоги до шкільного базового комплексу програмного забезпечення:

- ❖ системне програмне забезпечення :
 - операційні системи;
 - операційні оболонки;
 - мережне програмне забезпечення;
 - антивірусні засоби;
 - засоби резервного копіювання і поновлення інформації;
 - інші утиліти, що розширюють можливості системи.
- ❖ програмне забезпечення загального користування:
 - текстові процесори;
 - системи комп'ютерної графіки;
 - електронні таблиці;
 - бази даних;
 - системи роботи з комп'ютерними телекомунікаціями.
 - інструментальні програмні засоби загального користування;
- ❖ програмне забезпечення підтримки видавничої діяльності для потреб школи.
- ❖ програмні засоби навчального призначення, в тому числі педагогічні програмні засоби;
Педагогічні програмні засоби (ППЗ) дозволяють забезпечити:
 - індивідуалізацію і диференціацію процесу навчання за рахунок поетапного досягнення мети;

- здійснення контролю зі зворотнім зв'язком, діагностикою та оцінкою результатів;
- самоконтроль та самокорекцію;
- самопідготовку і тренаж;
- наочність;
- моделювання та імітацію певних явищ та процесів;
- проведення лабораторних робіт в режимі, спряженому з об'єктом;
- створення інформаційних баз даних, необхідних у навчальній діяльності, забезпечення доступу до інформаційних мереж;

- підсилення мотивації навчання;
- формування логічного мислення;
- розвиток творчих здібностей.

Педагогічні програмні засоби мають задовольняти такі дидактичні **вимоги**:

- ◆ науковість змісту матеріалу;
- ◆ відповідність матеріалу раніше здобутим знанням, вмінням і навичкам;
- ◆ пристосовуваність до індивідуальних можливостей учнів;
- ◆ систематичність і послідовність навчання;
- ◆ наочність;
- ◆ свідомість і активність дій учнів;
- ◆ міцність засвоєння матеріалу.

Види ППЗ:

- програми-тренажери, призначені для самопідготовки;
- контролюючі програми;
- демонстраційні програми;
- інформаційно-довідкові програми;
- імітаційно-моделюючі програми;
- ігрові програми;
- навчаючі програми.

Етапи розробки педагогічних програмних засобів:

- ◆ визначення мети і задачі ППЗ;
- ◆ вибір відповідного типу ППЗ;
- ◆ підбір фактичного матеріалу;
- ◆ систематизація і алгоритмізація матеріалу;
- ◆ програмне втілення;
- ◆ тестування програми і її корекція;
- ◆ проведення педагогічного експерименту з ППЗ.

2. Завдання до лабораторної роботи

1. Опрацювати відповідний матеріал шкільного підручника або навчального посібника.
2. Ознайомитися з основним примірним програмним забезпеченням шкільного курсу інформатики для шкіл, що працюють під управлінням операційних систем Windows та DOS:

Програмне забезпечення	OC MS-DOS	OC Windows
Клавіатурний тренажер	KEY.EXE	WCORRECT.EXE
Текстовий редактор	Модель текстового редактора – VIEW.EXE	Текстовий процесор – MS WORD
Графічний редактор	Модель графічного редактора – EDGR.EXE	Графічний редактор – MS PAINT
Електронні таблиці	Модель електронних таблиць – SHEET.EXE	Електронні таблиці – MS EXCEL
СУБД	Модель СУБД – BASE.EXE	СУБД - MS ACCESS
Інформаційно-пошукова система	Навчальна система – BTRAN.EXE	
Алгоритмічні ігри	Гра Баше – BASHE.EXE, Ханойська башта – HANOJ.EXE	
Навчальна алгоритмічна мова	Виконавець Черепаха – TURTLE.EXE	
Педагогічні програмні	Програма з геометрії –	

засоби	GEOMETR.EXE Програма з алгебри – GRAFX.Y.EXE) Програма з фізики – TRAEK.EXE) Програма з хімії – TABLE.EXE) Програма з біології – DIGIB.EXE)	
--------	---	--

3. Скласти програму, відповідно до варіанту (див. Додаток) з допомогою виконавця Черепаха.
4. Дати відповідь на контрольні питання.
5. Оформити звіт по лабораторній роботі.

Контрольні питання.

1. Чи можна виграти у комп'ютера в гру Баше при числі предметів 17 і максимальній їх кількості, яку можна взяти за один хід – 3? Якщо так, то як?
2. Яке число ходів у грі "Ханойська башта" при кількості дисків 64?
3. Скільки часу знадобиться на розв'язування задачі, якщо за 1 с робити 1 мільйон ходів ?

Додаток

1. Скласти програму, що обраховує кількість цифр в десятковому записі цілого числа $n > 0$;
2. Скласти програму, що обраховує старшу цифру в десятковому записі цілого числа $n > 0$;
3. Скласти програму, що обраховує найменшу цифру в десятковому записі цілого числа $n > 0$;
4. Скласти програму, що обраховує відстань від точки на числовій прямій з координатою x до найближчої точки відрізка $[0; 1]$;
5. Скласти програму, що обраховує відстань від точки $(x; y)$ на площині до найближчої точки відрізка $[0; 1]$ на осі Ox ;
6. Скласти програму, що обраховує закономірність і продовження вправо такої послідовності 1, 2, 6, 24, 120, 720, 5040, ...
7. Скласти програму, що обраховує закономірність і продовження вправо такої послідовності 1, 1, 0, 1, -1, 2, -3, 5, -8, 13, -21, ...
8. Скласти програму, що обраховує відстань від точки $(x; y)$ на площині до найближчої точки одиничного кола з центром в початку координат;
9. Скласти програму, що обраховує суму $1!+2!+3!+\dots+n!$;
10. Скласти програму, що обраховує кількість максимальних серед трьох чисел a, b, c ;
11. Скласти програму, що обраховує кількість різних серед трьох чисел a, b, c ;
12. Скласти програму, що обраховує середнє за величиною серед трьох чисел a, b, c ;
13. Скласти програму, що обраховує найменший дільник цілого числа $n > 0$, відмінний від 1;
14. Скласти програму, що сортує три числа a, b, c за величиною;
15. Скласти програму, що сортує три числа a, b, c за кількістю цифр цілої частини;
16. Скласти програму, що сортує три числа a, b, c за кількістю цифр дробової частини;
17. Скласти програму, що обраховує найбільший спільний дільник цілих чисел a і b ;
18. Скласти програму, що обраховує найменше спільне кратне цілих чисел a і b ;
19. Скласти програму, що обраховує кількість цілих серед чисел a, b, c, d ;
20. Скласти програму, що обраховує кількість дробових серед чисел a, b, c, d ;

Методика вивчення арифметичних, логічних та фізичних основ ЕОМ

ОСНОВНА
МЕТА

Розгляд основних особливостей вивчення арифметичних, логічних та логічних основ комп'ютера.

ВИМОГИ ДО ЗНАНЬ І
ВМІНЬ

Знати правила алгебри логіки. Вміти виконувати дії над булевими змінними. Розуміти сутність виконання логічних операцій. Знати термінологію. Вміти аналізувати роботу логічних схем.

1. Теоретичні відомості та методичні рекомендації

Почати вивчення матеріалу доцільно почати з поняття інформації. Зокрема слід наголосити, що у процесі науково-практичної діяльності людство виділило такі узагальнені абстрактні поняття, як речовина (матерія), енергія та інформація.

Термін “інформація” походить від латинського *informatio*, що означає пояснення, виклад, тлумачення. Поняття “інформація” багатозначне, належить до первісних понять інформатики. Строгого означення йому не дають, але характеризують його, перераховуючи властивості інформації.

Властивості інформації:

- 1) *Об'єктивність*. Інформація – відображення зовнішнього світу, а він існує незалежно від нашої свідомості, знань, думок та суджень про нього;
- 2) *Достовірність*. Інформація є достовірною, якщо вона відображає істинний стан справ. На основі достовірної інформації можуть прийматися правильні рішення;
- 3) *Повнота*. Інформацію можна назвати повною, якщо її достатньо для розуміння ситуації та прийняття рішення;
- 4) *Актуальність* інформації – важливість, істотність для даного часу;
- 5) *Корисність*. Ступінь корисності інформації залежить від потреб конкретних людей і від тих задач, які за допомогою отриманої інформації можна розв'язати.

Таким чином, *інформація* – це набір відомостей про об'єкти, явища і процеси навколишнього світу. *Інформація* – це відомості, знання, які ми отримуємо із зовнішнього світу або які передаються від одного об'єкта до іншого (наприклад, від одного комп'ютера до іншого).

Характерні риси інформації:

- 1) Інформація – це нематеріальна субстанція, але передається вона за допомогою матеріальних носіїв – знаків і сигналів – або за допомогою фізичних процесів, які змінюються з перебігом часу;
- 2) Інформація залежить не тільки від самих знаків та сигналів, але від їх взаємного розташування;
- 3) Якщо з однієї точки простору інформація передається в іншу, то в початковій точці вона не зникає;
- 4) Інформація є зрозумілою лише для того, хто здатний її розпізнати.

У широкому розумінні *інформація* – це відображення реального (матеріального світу) у вигляді знаків та сигналів.

Інформація, що зберігається, передається та опрацьовується в різних системах, є об'єктом вивчення інформатики.

Для передавання, зберігання і опрацювання інформації використовують *код* – систему умовних знаків (символів, літер).

Двійкові коди – спосіб подання інформації за допомогою двох символів – 0 і 1 (наприклад, число 34 у двійковій системі числення буде мати вигляд 100 010). Такий спосіб кодування обумовлено тим, що в пристроях комп'ютера використовуються елементи, які мають два відмінні стани (які називають 0 і 1). Це технічно легко реалізує зберігання і опрацювання інформації.

Біт (від англ. bit (аббревіатура [b]inary dig[it]) – двійкова одиниця) – це найменша одиниця вимірювання ємності пам'яті комп'ютера. Біт – це одна двійкова чарунка пам'яті, в яку можна записати 0 або 1.

Байт – це вісім підряд записаних бітів. Загальна кількість різноманітних комбінацій двійкових значень у байтові дорівнює 256.

Використовуються також похідні одиниці виміру ємності пам'яті комп'ютерів: кілобайт (Кб), мегабайт (Мб), гігабайт (Гб) тощо.

1 Кб = 2^{10} байт = 1024 байт;

1 Мб = 2^{20} байт = 1024 Кб;

1 Гб = 2^{30} байт = 1024 Мб.

Кожному байту присвоюють певний знак, отримуючи при цьому різноманітні коди.

ASCII код (від англ. American standard code for information interchange – американський стандартний код для обміну інформацією) – широко розповсюджена схема кодування знаків.

Крім арифметичних виразів використовують і логічні.

Логічний вираз – це засіб запису умов для пошуку інформації, її аналізу і відбору даних.

Перш за все слід нагадати учням відомості як відображається двійкова інформація у вигляді електричних сигналів; які існують пристрої для перетворення цих сигналів; як працюють логічні пристрої І, АБО, НЕ.

Слід нагадати таблиці істинності логічних висловлювань:

$y = a \text{ I } b$	$y = a \text{ АБО } b$	$y = \text{НЕ } a$
0 I 0 = 0	0 АБО 0 = 0	НЕ 0 = 1

$$0 \text{ І } 1 = 0 \quad 0 \text{ АБО } 1 = 1 \quad \text{НЕ } 1 = 0$$

$$1 \text{ І } 0 = 0 \quad 1 \text{ АБО } 0 = 1$$

$$1 \text{ І } 1 = 1 \quad 1 \text{ АБО } 1 = 1$$

та основні закони алгебри логіки:

$$a \text{ І } 1 = a \quad a \text{ І } 0 = 0 \quad a \text{ І } a = a$$

$$a \text{ АБО } 1 = 1 \quad a \text{ АБО } 0 = a \quad a \text{ АБО } a = a$$

$$a \text{ АБО } a \text{ І } b = a$$

$$a \text{ І } (a \text{ АБО } b) = a$$

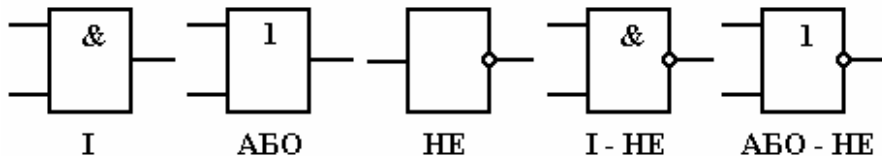
$$a \text{ І } b = \text{НЕ} (\text{НЕ } a \text{ АБО } \text{НЕ } b)$$

$$a \text{ АБО } b = \text{НЕ} (\text{НЕ } a \text{ І } \text{НЕ } b)$$

$$\text{НЕ } \text{НЕ } a = a$$

$$a \text{ І } (b \text{ АБО } c) = a \text{ І } b \text{ АБО } a \text{ І } c$$

Після цього слід нагадати умовні позначення логічних елементів на принципових схемах:



$$\text{І} : \text{Вихід} = \text{Вхід } 1 \text{ І } \text{Вхід } 2$$

$$\text{АБО} : \text{Вихід} = \text{Вхід } 1 \text{ АБО } \text{Вхід } 2$$

$$\text{НЕ} : \text{Вихід} = \text{НЕ } \text{Вхід}$$

$$\text{І-НЕ} : \text{Вихід} = \text{НЕ } (\text{Вхід } 1 \text{ І } \text{Вхід } 2)$$

$$\text{АБО-НЕ} : \text{Вихід} = \text{НЕ } (\text{Вхід } 1 \text{ АБО } \text{Вхід } 2)$$

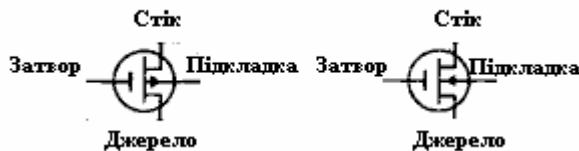
Логічні елементи можуть мати по кілька входів. При цьому їх дії підпорядковуються законам алгебри логіки:

$$a \text{ І } b \text{ І } c \text{ І } d = (a \text{ І } b) \text{ І } (c \text{ І } d);$$

$$a \text{ АБО } b \text{ АБО } c \text{ АБО } d = (a \text{ АБО } b) \text{ АБО } (c \text{ АБО } d).$$

Інформація зберігається й обробляється в ЕОМ у двійковому вигляді — у вигляді послідовностей нулів і одиниць. У сучасних ЕОМ широко поширене представлення "1" високою напругою (наприклад, +5 Вольт), а "0" — низькою (близько 0 Вольт).

Основним елементом сучасних ЕОМ є транзистор. В обчислювальній техніці транзистори використовуються в так званому *ключовому* режимі. У цьому режимі транзистор можна представити як звичайний вимикач, що в одному положенні проводить струм (замкнутий), а в іншому — ні (розімкнутий).

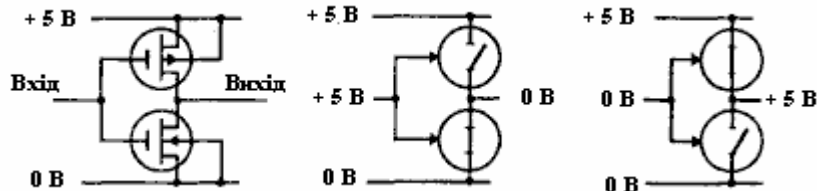


Мал. 1

Мал. 2

Розглянемо для прикладу *МОП-транзистор* (мал. 1; скорочення "МОП" — Метал-Оксид-Напівпровідник — відноситься до будови транзистора). Кожен такий транзистор має чотири контакти: стік, джерело, затвор і підкладку. N-канальний транзистор (мал. 2) проводить струм між джерелом і стоком (замкнутий), тільки якщо на затвор подана позитивна (щодо підкладки) напруга. P-канальний транзистор, навпаки, проводить струм тільки якщо напруга на затворі (мал. 1) менше напруги на підкладці.

У якості елементарних будівельних "кубиків" в ЕОМ звичайно використовуються не самі транзистори, а більш великі елементи, що називаються *вентиллями*. Існують різні способи (технології) об'єднання транзисторів у вентилі.



Мал. 3

Мал. 4

Мал. 5

Розглянемо для прикладу вентиль "не" (мал. 3). Якщо на його вхід подати напругу +5 В ("1"), то нижній транзистор замкнеться, а верхній буде розімкнутий (мал. 4) — на виході вентилля буде напруга 0 В ("0").

Якщо ж на вхід вентилля подати 0 В, то замкнеться верхній транзистор, оскільки напруга на його затворі (0 В) буде менше напруги на підкладці (+5 В). На нижньому ж транзисторі напруга на затворі (0 В) і підкладці (0 В) будуть збігатися і транзистор буде розімкнутий (мал. 5). У результаті на виході вентилля буде +5 В ("1").

Таким чином, розглянутий вентиль є найпростішим електронним пристроєм обробки інформації — він інвертує ("перевертає") біт: "1" (+5 В) на вході перетворює в "0" (0 В) на виході, а "0" перетворює в "1". Якщо поставити "1" значення

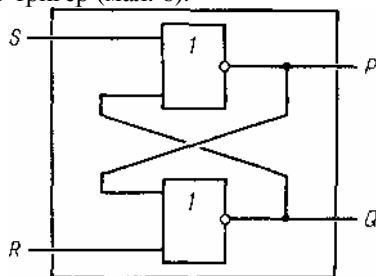
так, а "0" - ні, то роботу цього вентиля можна описати формулою

вихід = **не** вхід

Тому розглянутий вентиль називається інвертором або вентилям "не".

Покажемо тепер, як можна інформацію запам'ятовувати. Оскільки будь-яка інформація в ЕОМ представляється в двійковому вигляді, розглянемо запам'ятовування і збереження елементарної порції інформації — одного біта. Електронна схема, що запам'ятовує один біт інформації, називається **тригером**.

Розглянемо найпростіший так називаний RS-тригер (мал. 6).



Мал. 6

Якщо на входи цього тригера подати $S=1$, $R=0$, то на виходах Q та P з'являться відповідно 1 та 0. Якщо тепер перестати подавати сигнали на тригер ($S=0$, $R=0$), то виходи P та Q не змінять свого значення. Таким чином, встановлені значення виходів P та Q не зміняться при переході до $S=0$, $R=0$.

Точно так само при подачі $S=0$, $R=1$ на виходах з'являться $Q=0$, $P=1$ і ці значення виходів збережуться при знятті "1" із входу R ($R=0$, $S=0$).

Таким чином, при $S=0$, $R=0$ тригер може знаходитися в двох різних станах: $Q=1$ і $Q=0$. Вихід Q і є значенням біта, що запам'ятався.

Оскільки один тригер запам'ятовує один біт, то для запам'ятовування байта (8 біт) потрібно 8 тригерів. Сучасні мікросхеми пам'яті обсягом менш 1 см^3 здатні запам'ятовувати мільйони біт інформації.

На уроках доцільно використати педагогічний програмний засіб "Логічні елементи". Даний ППЗ є типовим прикладом програм-тренажерів, призначених для ретельного опрацювання деякої вузької теми або розділу курсу. Ніякої додаткової інформації для роботи з програмою не вимагається, тому, пригадавши спочатку відповідний теоретичний матеріал, можна одразу приступати до виконання вправ.

В демонстраційній частині програми учневі надається можливість розглянути принципи дії логічних елементів, які виконують логічні операції "І", "АБО", "НЕ".

В процесі роботи з демонстраційною частиною програми учень повинен встановити на схемі значення вхідних сигналів A і B у різноманітних комбінаціях шляхом послідовного відкриття і закриття вентилів, положення яких відповідає певним значенням вхідних сигналів таким чином:

вентиль закритий – значення вхідного сигналу 0;

вентиль відкритий – значення вхідного сигналу 1.

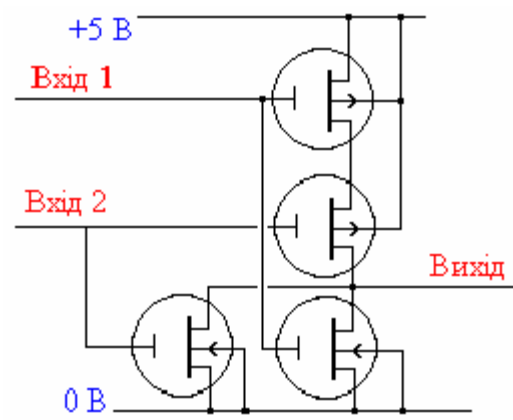
В таблиці дійсності кожний раз висвічується відповідний рядок, який відображає конкретну ситуацію, встановлену на схемі.

2. Завдання до лабораторної роботи

- Опрацювати матеріал §2 та §§5-10 глави 1 шкільного навчального посібника [5] та §17 шкільного навчального посібника [8].
- Опрацювати педагогічний програмний засіб "Логічні елементи" (Logel.exe).
- Скласти конспект уроку по темі "Арифметичні та логічні основи ЕОМ", врахувавши методичні рекомендації.
- Дати відповідь на контрольні питання.
- Оформити звіт по лабораторній роботі.

Контрольні питання.

- Чому дорівнює: $a \text{ І } b \text{ І } c$, якщо $a=1$, $b=1$, $c=0$?
- Чому дорівнює: $a \text{ І } c \text{ АБО } a \text{ І } b \text{ АБО } d \text{ І НЕ } d$, якщо $a=1$, $b=0$, $c=1$, $d=0$?
- Яка напруга буде на виході вентиля, зображеного на малюнку 3, якщо на його вхід подати напругу +5 В? 0 В?
- Яка напруга буде на виході вентиля, зображеного на малюнку 7, якщо на перший вхід подати напругу +5 В, а на другий - напругу 0 В?
- Що буде на виходах P та Q верхнього вентиля, зображеного на малюнку 6, якщо на входи цього тригера подати $S=1$, $R=0$? $S=0$, $R=1$?



Мал. 7

ЛАБОРАТОРНА РОБОТА №5 (4 год.)

Методика вивчення інформаційної системи

ОСНОВНА
МЕТА

Методика вивчення структури інформаційної системи в школі.

ВИМОГИ ДО ЗНАНЬ І
ВМІНЬ

Знати основні складові частини комп'ютера, їх призначення та принципи роботи. Вміти пояснити відповідний матеріал учням загальноосвітніх шкіл.

1. Теоретичні відомості з інформатики до теми „Інформаційна система”

Основою будь-якої інформаційної системи є обчислювальна машина (комп'ютер).

Під структурною організацією ЕОМ розуміють їх деяку фізичну модель, яка відображає склад, порядок і принципи основних функціональних модулів без деталей їх технічної реалізації.

Аналізуючи будову комп'ютера на логічному рівні, у структурі комп'ютера виділяють центральні пристрої та пристрої введення-виведення (зовнішні пристрої).

До центральних пристроїв відносять мікропроцесор та пам'ять. Зовнішні пристрої часто називають периферійними.

Ядром обчислювального процесу в комп'ютері є його мікропроцесор.

Мікропроцесор (МП) – це мікросхема (надвелика інтегральна схема), яка керує роботою всіх апаратних компонентів комп'ютера, забезпечує виконання програм, виконує арифметичні та логічні операції.

Процесор має дві основні характеристики – швидкодія (тактова частота) та розрядність.

Швидкодія – це кількість операцій за 1 секунду.

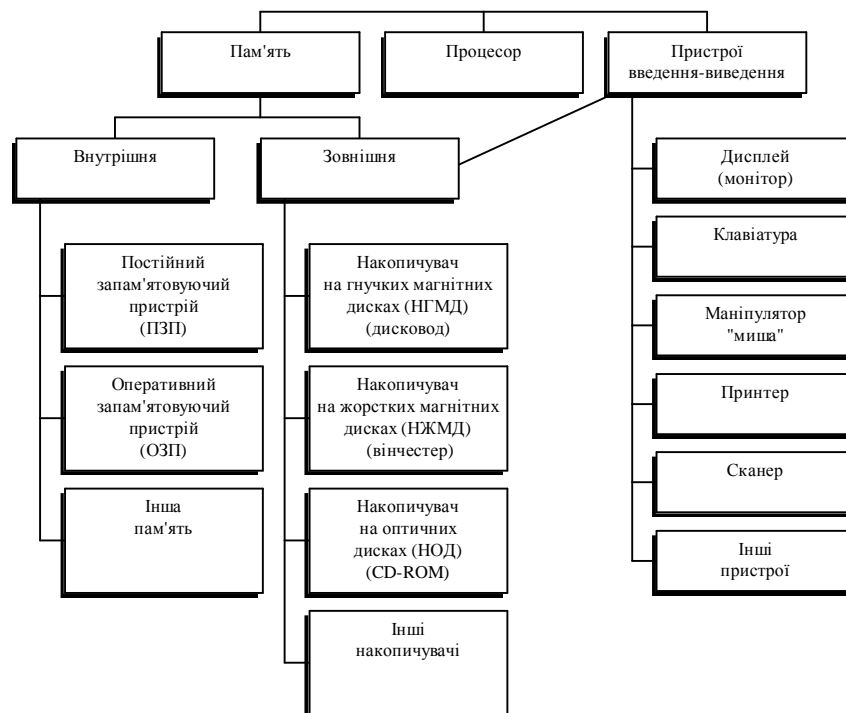
Розрядність – кількість інформації, що обробляється за 1 операцію.

Пам'ять поділяється на зовнішню та внутрішню.

До внутрішньої пам'яті відносять оперативну (оперативний запам'ятовуючий пристрій – ОЗП), постійну (ПЗП), відсопам'ять та інші види пам'яті.

Оперативна пам'ять (ОЗП) призначена для зберігання даних та програм протягом одного сеансу роботи з комп'ютером (від моменту включення комп'ютера до моменту його виключення). В ОЗП інформацію можна заносити та читати з неї. У ній зберігаються дані, з якими ведеться оперативна робота, та програми для їх обробки.

У постійній пам'яті (ПЗП) містяться програми, які забезпечують функціонування обчислювальної системи (зокрема, програми початкового завантаження комп'ютера та тестування його складових) та деякі програми з базового програмного забезпечення комп'ютера. Цю пам'ять називають постійною тому, що інформація в ній зберігається після виключення комп'ютера. ПЗП “прошивається” інформацією на етапі її виготовлення. З постійної пам'яті інформацію можна лише читати, записати туди користувач нічого не може.



ОЗП і ПЗП утворюють основну пам'ять. Основна пам'ять комп'ютерів складається з окремих комірок. Усі комірки пам'яті нумеруються, починаючи з нуля; номери комірок пам'яті називаються їх адресами.

Технічною реалізацією запам'ятовуючих пристроїв для організації основної пам'яті є мікросхеми пам'яті.

Відеопам'ять – це пам'ять, що служить для підготовки та збереження зображення на екрані.

Кеш-пам'ять – це невелика за розміром буферна пам'ять, яка дозволяє пришвидшувати процеси обміну даними між МП і ОЗП за рахунок значно меншого, ніж до ОЗП, часу доступу до даних та розташування в ній даних, які найчастіше використовуються.

До основних зовнішніх пристроїв належать:

1) *клавіатура* – пристрій, призначений для введення до ПК символічної інформації (літер, цифр, розділових знаків тощо), а також для управління роботою ПК. Для кожного символу виділяється клавіша, натискаючи на яку, ми і вводимо код символу в ПК.

2) *монітор* – пристрій для відображення на екрані текстової та графічної інформації. До основних характеристик дисплею відносять: роздільну здатність (кількістю точок по вертикалі та горизонталі екрану. Чим більша кількість точок, тим вища роздільна здатність, тобто вища якість зображення), глибина кольору – кількість бітів, яка необхідна для запам'ятовування коду кольору однієї точки, кількість кольорів, розмір екрану (вимірюють по діагоналі в дюймах (2,54см). Найпоширенішими у використанні є 14-дюймові екрани, але використовують також 15-, 17-, 19-, 20-, 21-дюймові екрани).

3) *ручний маніпулятор* “миша” – для вибору тих чи інших параметрів, режимів роботи, команд програм та для керування роботою рухомих об'єктів на екрані монітора;

4) *накопичувачі на гнучких магнітних дисках* (дискети або флоппі-диски). Найчастіше використовують 3,5-дюймові дискети з записами інформації на обох поверхнях високої щільності (HD) - 1,44 Мбайт. Нову дискету перед використанням необхідно підготувати до роботи - відформатувати. Форматування магнітного накопичувача — це процес його розбивання на сектори та доріжки засобами операційної системи. Більшість дискет продаються відформатованими виробником (на упаковках або на дискеті нанесені мітки “DOS formatted” або “IBM formatted”). Для зчитування або запису інформації на дискети ПК комплектуються спеціальними пристроями – дисководами.

5) *накопичувачі на жорстких магнітних дисках* (“вінчестери”). Вінчестери, порівняно з дискетами, зберігають значно більші обсяги інформації та мають на порядок вищу швидкість запису та зчитування інформації. Вінчестери мають місткість від 4 Гбайт і більше. Вінчестери, на відміну від дискет, не є знімними, але є можливість знімати вінчестер з одного комп'ютера і підключати до іншого. Вінчестери, як правило, форматуються продавцями ПК.

6) *накопичувачі на оптичних дисках* (компакт-диски - CD). Такі диски випускають діаметром 5,25 та 3,5 дм. Місткість пам'яті CD у декілька сот разів вища від місткості магнітних дискет – від 700 Мбайт. Розрізняють щонайменше два види компакт-дисків: компакт-диски тільки для читання (CD-ROM) та записувані компакт-диски, що дозволяють не тільки зчитувати, а й записувати інформацію (CD-R – для одноразового запису, CD-RW – для багаторазового запису). CD можна використовувати лише за наявності відповідного дисковода – пристрою для читання та запису інформації на компакт-диски. Накопичувачі на дисках є одночасно видами зовнішньої пам'яті.

7) *принтер* – пристрій друкування, призначений для виведення на папір, а в деяких випадках - і на спеціальну плівку символічної та графічної інформації. Найбільш вживані типи принтерів: матричні, струминні та лазерні. Будь-який принтер має дві характеристики: швидкість друкування (сторінок за хвилину) та якість. Якість друкування визначається роздільною здатністю принтера, яка виражається кількістю друкованих крапок на дюйм (dpi). Найбільш розповсюдженими у використанні є принтери матричного типу. Спосіб одержання зображення символів у них подібний до способу одержання зображення символів на екрані дисплею. Друкувальна голівка матричного принтеру має металеві голки, які, одержавши сигнал від комп'ютера, ударяють по фарбувальній стрічці і залишають на папері відмітки — крапки, що утворюють зображення. У струминних принтерах зображення одержується шляхом нанесення на папір рідкого чорнила. Через спеціальні форсунки тонер виплискується на папір. Швидкість та якість друкування цих принтерів краща, ніж у матричних. Струминні принтери дозволяють здійснювати як чорно-білий, так і кольоровий друк. Лазерні принтери забезпечують найвищу швидкість і найкращу якість, порівняно з описаними вище типами принтерів. Їх принцип дії такий: лазерний промінь електризує поверхню друкуючого світлочутливого барабану, ділянки барабану зі зміненою полярністю притягують спеціальний порошок, який потім наноситься на папір; далі аркуш паперу з нанесеним порошком піддається термічній обробці. Друкування цих принтерів може бути чорно-білим і кольоровим.

8) *сканер* – пристрій, призначений для введення в комп'ютер графічних об'єктів (малюнків, фотографій та ін.). Сканери можуть використовуватися і для введення текстової інформації. При цьому повинна бути використана спеціальна програма, яка здійснює розпізнавання графічних зображень символів і їх перетворення в коди формату ASCII. Основною характеристикою сканера є роздільна здатність, яка визначається кількістю точок на дюйм. Зараз використовують сканери з роздільною здатністю 300-2400 dpi. Сканери поділяють на ручні, планшетні, роликові та проекційні.

9) *плотер* – пристрій для побудови графічних зображень на папері;

10) *стрімер* – пристрій для збереження інформації на магнітних стрічках;

11) *модем* – пристрій для підключення комп'ютерів до мереж;

12) *мультимедійне обладнання* надає можливість використовувати на комп'ютері різні види інформації, а саме: відео і звукову. Мінімальний комплект мультимедійного обладнання включає: звукові плати, які розміщуються в системному блоці, обладнання CD-ROM і звукові динаміки. Це обладнання дозволяє зображення на екрані супроводжувати звуком (музикою, дикторською мовою тощо).

Для узгодження роботи центральних та зовнішніх пристроїв використовуються *інтерфейсні блоки*. Вони здійснюють перетворення сигналів, які передаються через системну магістраль, в сигнали, які забезпечують роботу відповідного зовнішнього пристрою, здійснюють буферизацію даних. Технічною реалізацією інтерфейсних блоків є контролер, адаптери, відеоплати, синтезатори звуку тощо.

Системна магістраль – це сукупність шин для обміну даними, адресами та керуючими сигналами між центральними та зовнішніми пристроями. Кожна шина – це деяка кількість паралельних ліній (ця кількість визначає

розрядність шини), по яких у вигляді електричних сигналів передається інформація від одного пристрою до іншого.

2. Теоретичні відомості з загальної методики навчання інформатики, що використовуються в лабораторній роботі

Форми, способи і засоби контролю й оцінки знань і умінь учнів. Норми оцінок.

Контроль — це виявлення і порівняння результату навчальної діяльності з вимогами, що задаються до цього результату програмою, тобто це – співвіднесення досягнутих результатів із запланованими цілями навчання.

У залежності від того, хто саме здійснює контроль за результатами діяльності учня, виділяють наступні три **типи** контролю:

- зовнішній (здійснюється вчителем над діяльністю учня);
- взаємний (здійснюється учнем над діяльністю товариша);
- самоконтроль (здійснюється учнем за власною діяльністю).

Основна **мета** контролю знань і умінь полягає у виявленні досягнень, успіхів учнів, через призму яких розглядаються недоліки в здійсненні навчальної діяльності, пробіли в знаннях і т.п.; у вказівці шляхів удосконалювання, поглиблення знань, умінь, для того щоб створювалися умови для наступного включення школярів в активну творчу діяльність.

Конкретизація цієї мети зв'язана з вирішенням таких **завдань**:

- установленням якості засвоєння учнями матеріалу, передбаченого програмою з інформатики для загальноосвітньої школи (установлення повноти, характеру виконання учнями завдань учителя);
- виявлення відповідності досягнутого школярами рівня оволодіння матеріалом прийнятим нормам;
- визначенням заходів коректування знань і умінь учнів;
- оволодіння школярами прийомами взаємоконтролю і самоконтролю, формуванням потреби в самоконтролі;
- вихованням ряду якостей особистості, наприклад відповідальності за виконувану роботу, ініціативи й ін.

Контроль у навчанні виконує наступні **функції**:

- виявлення і діагностика результатів навчання (контролююча і діагностична функції);
- освітню (навчальну), пов'язану з підвищенням якості засвоєння знань, їхньою систематизацією, формуванням прийомів навчальної роботи;
- стимулюючу (розвиваючу), пов'язану зі створенням необхідної основи для стимулюючих змістовних оцінок діяльності учнів, для розвитку пізнавальної активності школярів;
- виховну, спрямовану на виховання в кожного почуття відповідальності за результати навчання, на формування пізнавальної мотивації;
- керування процесом засвоєння знань, умінь, його корекції (прогностична функція).

У процесі контролю знань і умінь учнів виділяються наступні **компоненти**:

- уточнення цілей вивчення даного відрізка навчального матеріалу і встановлення конкретного змісту контролю;
- вибір видів, форм, способів і засобів контролю, що відповідають поставленим цілям;
- різні способи вираження результатів контролю: оцінка й відмітка.

Види контролю:

- контроль за кінцевим результатом;
- покроковий контроль;
- контроль, пов'язаний з установленням визначених параметрів діяльності;
- поточний;
- тематичний;
- підсумковий за курсом навчання.

Очевидно, з погляду навчального ефекту переважає покроковий контроль, тому що в його процесі учень усвідомлює сутність і характер діяльності.

Форми контролю знань і умінь учнів виділяються відповідно до форм навчання:

- масова (іноді в ній виділяють групову і фронтальну);
- індивідуальна.

Можна вказати і конкретні форми, що можуть бути віднесені як до масових, так і до індивідуальних. Це – залік, фронтальне, індивідуальне опитування, контрольні роботи, твори, диктанти.

Виділяють різні **способи** контролю знань і умінь школярів: письмовий, усний, практичний (зв'язаний з виконанням різного роду лабораторних і практичних робіт).

Говорячи про **засоби** контролю знань і умінь, найчастіше мають на увазі завдання або кілька завдань, що пропонуються учням з метою виявлення відповідно до поставлених цілей результатів навчання.

В основу класифікації таких засобів може бути покладена форма введення відповіді на контролююче завдання.

У цьому випадку виділяються:

- *завдання вільного вибору відповіді*;
- *тести* (введення відповіді певним чином обмежується).

Завдання вільного вибору передбачають вільне конструювання відповіді учнів. Такі завдання в залежності від характеру навчально-пізнавальної діяльності учнів при їхньому виконанні можуть бути розділені на:

- *питання* (в основі – діяльність відтворення);
- *задачі* (виконання цих завдань припускає сформованість дій, що складають основу діяльності по розв'язуванню задачі).

Тести поділяються на два види: *тести на пригадування і доповнення, вибіркові*.

Тести першого виду являють собою завдання учням заповнити пропуски в запропонованому їм зв'язному тексті (наприклад, зошит із друкованою основою).

Вибіркові тести поділяються на *альтернативні, перехресного вибору і множинного вибору*.

Альтернативний тест — це завдання, виконавши яке учень із двох запропонованих йому відповідей повинний вибрати одну (на його думку, правильну).

Тест перехресного вибору являє собою кілька завдань, після виконання яких учень встановлює відповідність отриманих їм результатів передбачуваним результатам, записаним у довільному порядку (число завдань і число запропонованих учням відповідей збігаються).

Тест множинного вибору складається з завдання і списку відповідей (серед відповідей — одна правильна). Учень повинний вибрати з цього списку ту відповідь, що, на його думку, є правильною.

Всі види тестів можливо реалізувати з допомогою комп'ютерних технологій. При цьому вчитель повинен дотримуватися таких вимог:

- 1) зміст завдання повинен відповідати меті перевірки;
- 2) однозначність поставленого завдання (особливо у тестах на пригадування і доповнення);
- 3) тести повинні містити максимум інформації про об'єкт контролю і в той час бути лаконічними;
- 4) завдання повинні супроводжуватися поясненнями для правильного їх опрацювання;
- 5) слід забезпечувати різноманітність тестових завдань для кожного учня;
- 6) в тестах множинного вибору слід обмежуватися 3-5 відповідями;
- 7) неправильні відповіді тестів повинні бути максимально ймовірними;
- 8) слід диференційовано підходити до оцінювання відповідей в залежності від складності завдання шляхом бальних критеріїв;
- 9) по можливості текстові завдання ілюструвати різноманітною наочністю (рисунок, таблиці, відеофайли тощо), що зменшує втому учнів і підвищує інтерес до роботи.

Процес контролю знань і умінь учнів пов'язаний з оцінкою й відміткою. Ці поняття варто розрізняти. Оцінка — це процес, дія (діяльність) оцінювання, що здійснюється людиною. Відмітка виступає як результат цього процесу (результат дії), як його умовне формальне вираження.

Оцінка й відмітка визначаються знаннями й уміннями учня, що він показав у процесі контролю. Одним з показників, по якому вчитель має можливість судити про ці знання, уміння, служать погіршеності, допущені учнями при роботі з засобами контролю, запропонованими вчителем.

Погіршеності поділяють на помилки і недоліки.

Помилка — це погіршеність, що свідчить про те, що учень не опанував тими знаннями й уміннями (зв'язаними з контрольованим розділом, темою), що визначені програмою з інформатики.

Недоліком вважають погіршеність, що вказує або на недостатньо повне, міцне засвоєння основних знань і умінь, або на відсутність знань, що програмою не відносяться до основних. До недоліків відносять також неакуратний запис, недбале виконання малюнка або оформлення розв'язку задачі і т.д.

Приведений розподіл погіршеностей на помилки і недоліки є умовним, і це потрібно мати на увазі в процесі оцінювання. Варто пам'ятати, що розмитість границі між помилкою і недоліком може бути однієї з причин необ'єктивної оцінки знань і умінь учня. У залежності від об'єкта контролю, від конкретних обставин погіршеність, що допущена учнем, може бути віднесена вчителем до розряду помилок або недоліків.

Як зазначалось, контроль виконує різноманітні функції: навчаючу, стимулюючу, виховну, діагностичну та інші. Однак, на жаль, в практиці навчання в переважній більшості переважають контролюючі функції перевірки, тобто виявлення стану знань і умінь учнів, рівня їх розумового розвитку та ступеня засвоєння прийомів пізнавальної діяльності, навичок раціональної навчальної праці, результатом чого є певна оцінка. Реалізуються ці задачі переважно шляхом проведення кількісного аналізу результатів навчання, що дозволяє встановити лише загальну картину, яка характеризує засвоєння учнями вивченого матеріалу.

Однак, як для учителя, так і для учня, важливішим є не просто знання про загальний рівень підготовки з тієї чи іншої теми, а отримання інформації про помилки, недоліки та пробіли в знаннях і вміннях, про причини такого стану та вплив цих причин на якість знань, умінь та навичок учня. Це дозволяє вибрати найбільш дієвий індивідуальний підхід, достатньо повну систему вправ, більш ефективну методику навчання і в результаті скласти прогноз проведення навчального процесу.

Реалізація діагностичної та прогностичної функцій контролю здійснюється шляхом проведення якісного аналізу результатів навчання. Проведення такого аналізу є трудомістким процесом і вимагає значного часу, що і пояснює його незначне застосування в практиці роботи учителів.

Що потрібно для правильного і ефективного проведення якісного аналізу результатів навчання? По-перше, підготовка відповідних контрольних завдань. Для цього потрібно виділити основні поняття, правила, формули, твердження, вміння та навички, що повинні бути засвоєні учнями і підлягають обов'язковому контролю. Далі необхідно підібрати або скласти відповідні питання, вправи, задачі, тести, що повністю охоплюють контролюючу базу і відповідають всім психолого-методичним вимогам, в тому числі багатоваріантності завдань. По-друге, самостійне виконання учнями запропонованих контрольних робіт. І, по-третє, безпосередня перевірка учителем контрольних завдань, при якій для кожного учня слід виділити допущені помилки і похибки, проектуючи їх на відповідний понятійний апарат, що контролюється.

Зрозуміло, що такий обсяг знань, умінь та навичок для контролю неможливо охопити 1-2 контрольними роботами. Трудомісткою є для учителя і повноцінна перевірка контрольних завдань з врахуванням критеріїв якісного аналізу.

Допомогти в цьому можуть нові інформаційні технології навчання, зокрема контролюючі педагогічні програмні засоби, які шляхом протоколювання відповідей учнів дозволяють ефективно провести якісний аналіз, що в свою чергу,

дозволяє учителю ціленаправлено підійти до підсумкових уроків теми, враховуючи потреби корекції знань, вмінь та навичок кожного учня, підняти ефективність процесу навчання в цілому.

Оцінювання навчальних досягнень учнів з інформатики здійснюється згідно з критеріями для підсумкового (тематичного) оцінювання навчальних досягнень.

В системі оцінювання навчальних досягнень учнів обов'язковим і основним є тематичне оцінювання знань, основною метою якого є забезпечення неупередженого оцінювання навчальних досягнень учнів з кожної теми. При цьому бал, отриманий учнем, має відображати реальні досягнення в опануванні ним конкретної теми.

Поточне оцінювання учнів з інформатики з виставленням балів до класного журналу, якщо воно здійснюється вчителем, може проводитись безпосередньо під час навчальних занять або за результатами виконання домашніх завдань, усних відповідей, лабораторних, практичних та самостійних робіт тощо.

Якщо темою передбачено виконання учнями практичних робіт, то це є *обов'язковою попередньою умовою* для тематичного оцінювання.

Підсумкове оцінювання здійснюється наприкінці кожного семестру та навчального року: за семестр проводиться за результатами тематичного оцінювання, а за рік - на основі семестрових оцінювань.

Для стимулювання навчальної діяльності учнів вчитель може за підсумками роботи учня при вивченні окремої теми автоматично оцінити відповідним балом його навчальні досягнення з даної теми (якщо учень погоджується з ним). При цьому наявність чи кількість поточних оцінок не може бути перешкодою для виставлення тематичної оцінки.

Якщо ж учень прагне підвищити бал за тему, то він має домовитися про час для цього зі вчителем. Зовсім не обов'язково, щоб це був залік, контрольна робота, тестове завдання. Вчитель і учні можуть це зробити в ході бесіди, факультативного заняття, на основі реферативної роботи тощо.

Критерії оцінювання навчальних досягнень учнів єдині для загальноосвітніх і профільних класів. Неприпустимо завищувати вимоги до виставлення відповідних балів під час контролю результатів навчання в класах з поглибленим вивченням інформатики чи, навпаки, занижувати їх у класах гуманітарного профілю, оскільки це негативно впливатиме на розвиток творчих здібностей учнів.

Для оцінювання навчальних досягнень учнів, які навчаються за безмашинним варіантом програми з інформатики, слід керуватися загальними критеріями оцінювання навчальних досягнень учнів без врахування навичок володіння комп'ютером.

Критерії оцінювання навчальних досягнень учнів з інформатики

Рівні навчальних досягнень учнів	Бали	Критерії навчальних досягнень учнів
Початковий	1	Учень з допомогою вчителя може розпізнати і назвати окремі поняття інформатики; стикається з непереборними труднощами під час виконання лабораторних, практичних та інших робіт.
	2	Учень з допомогою вчителя або підручника наводить елементарні приклади і поняття з інформатики; за інструкцією із допомогою вчителя частково виконує лабораторні та практичні роботи без оформлення.
	3	Учень з допомогою вчителя або підручника фрагментарно характеризує окремі поняття з інформатики; за інструкцією із допомогою вчителя виконує лабораторні та практичні роботи із частковим їх оформленням без висновків.
Середній	4	Учень з допомогою вчителя або підручника дає визначення окремих понять з інформатики, неповно характеризує загальні ознаки інформаційних об'єктів; за інструкцією із допомогою вчителя виконує лабораторні та практичні роботи з неповним їх оформленням.
	5	Учень самостійно дає визначення окремих понять з інформатики, з допомогою вчителя або підручника відтворює навчальний матеріал; характеризує загальні ознаки інформаційних об'єктів; за інструкцією виконує лабораторні та практичні роботи, звертаючись по консультацію до вчителя, оформляє їх, не зробивши висновків.
	6	Учень самостійно, але не повно відтворює навчальний матеріал; характеризує будову та функції окремих інформаційних об'єктів, наводить прості приклади; з допомогою вчителя розв'язує прості типові вправи з інформатики; за інструкцією виконує лабораторні та практичні роботи, оформляє їх, робить висновки, що не відповідають меті роботи.
Достатній	7	Учень самостійно відтворює навчальний матеріал, розкриває суть понять з інформатики; розв'язує прості типові вправи та задачі; за інструкцією виконує лабораторні та практичні роботи, оформляє їх, робить неповні висновки.
	8.	Учень самостійно відповідає на поставлені запитання; дає порівняльну характеристику об'єктів інформатики, розв'язує типові вправи та задачі, виправляє допущені помилки; за інструкцією виконує лабораторні та практичні роботи, оформляє їх, робить нечітко сформульовані висновки.
	9	Учень вільно відповідає на поставлені запитання; самостійно розв'язує вправи і задачі, виправляє помилки; з допомогою вчителя встановлює причинно-наслідкові зв'язки; виконує лабораторні та практичні роботи, оформляє їх, робить чітко сформульовані висновки.

Високий	10	Учень обґрунтовано відповідає на запитання, передбачені навчальною програмою; самостійно аналізує і розкриває суть понять, алгоритмів, програм, узагальнює, систематизує, встановлює причинно-наслідкові зв'язки; виконує лабораторні та практичні роботи, оформляє їх виконання, робить логічно побудовані висновки відповідно до мети роботи..
	11	Учень логічно, усвідомлено відтворює навчальний матеріал у межах програми; самостійно аналізує і розкриває закономірності інформаційних процесів, оцінює об'єкти інформатики, закони; встановлює і обґрунтовує причинно-наслідкові зв'язки; ретельно виконує лабораторні та практичні роботи, оформляє їх, робить обґрунтовані висновки.
	12	Учень виявляє міцні й глибокі знання з інформатики, може вести дискусію з конкретного питання з використанням міжпредметних зв'язків, самостійно оцінює та характеризує різноманітні об'єкти інформатики, виявляє, особисту позицію щодо них, уміє розв'язувати проблемні завдання; самостійно користується джерелами інформації, що рекомендовані вчителем; ретельно виконує лабораторні та практичні роботи, робить обґрунтовані висновки, справляється з додатковими завданнями.

3. Завдання до лабораторної роботи

- Опрацювати відповідний матеріал шкільного підручника або навчального посібника.
- Опрацювати матеріал загальної методики "Форми, способи і засоби контролю й оцінки знань і умінь учнів. Норми оцінок."
- Розробити систему перевірки знань та умінь учнів з даної теми, що охоплює **всі основні поняття та означення** та повинна включати:
 - систему контрольних питань для вільного вибору відповіді;
Наприклад. Які бувають накопичувачі на дисках?
 - систему контрольних питань для диктанту з інформатики;
Наприклад. Вставте пропущене слово у реченні: "Кількість команд, що їх виконує центральний процесор за одиницю часу – це ...". Відповідь – "швидкодія".
 - систему тестів множинного вибору.
Наприклад. Число бітів, що одночасно обробляються називається:
 - швидкодією процесора;
 - розрядністю процесора;
 - швидкістю процесора.
- Кожне контрольне питання і тест оцінити певною кількістю балів від 1 до 3.
- Обмінятися розробленими системами з іншим студентом. Відповідно виконати завдання та оцінити виконання товариша за критеріями²:
 - на оцінку "11" – 100% набраних балів;
 - на оцінку "10" – 95% – 99% балів;
 - на оцінку "9" – 85% – 94% балів;
 - на оцінку "8" – 80% – 84% балів;
 - на оцінку "7" – 75% – 79% балів;
 - на оцінку "6" – 65% – 74% балів;
 - на оцінку "5" – 55% – 64% балів;
 - на оцінку "4" – 45% – 54% балів;
 - на оцінку "3" – 30% – 44% балів;
 - на оцінку "2" – 20% – 29% балів;
 - на оцінку "1" – менше 20% балів.
- Оформити звіт по лабораторній роботі (представити систему контролю, оцінювання товариша за цією системою та його зауваження до розробленої системи).

² Відповідно до наказу № 341 МОІН України від 25.04.2001р.

Методика вивчення операційної системи MS DOS

ОСНОВНА МЕТА	Розгляд основних методичних особливостей вивчення операційної системи MS DOS в школі.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ	Знати основні принципи, форми, прийоми та засоби вивчення операційної системи MS DOS.
ППЗ	Зовнішні команди операційної системи MS DOS. Тестуюча система IQ Тест.

1. Теоретичні відомості та методичні рекомендації

Операційна система є посередником між ЕОМ та користувачем. Вона робить роботу з ЕОМ більш простою, звільняючи програміста від трудомісткої роботи, пов'язаної з розподілом ресурсів, керуванням апаратурою й організацією виконання програми.

Операційна система здійснює аналіз запитів користувача і забезпечує їх виконання. Запит відображає необхідні ресурси і потрібні дії ЕОМ і є послідовністю команд на особливій мові директив операційної системи. Така послідовність команд називається завданням.

Операційна система MS-DOS (дисконна операційна система) створена фірмою Microsoft і з'явилася в 1981 році практично одночасно з персональним комп'ютером фірми IBM.

В повному складі ОС ПК містить такі основні складові:

- драйвери зовнішніх пристроїв;
- файлова система;
- процесор вказівок операційної системи.

Сукупність драйверів стандартних пристроїв складає базову систему введення-виведення. Ці програми управляють операціями введення інформації від зовнішніх пристроїв і виведення на зовнішні пристрої.

Файлова система – це система програмних засобів, які забезпечують зберігання, використання і опрацювання файлів-програм, даних, текстів, зображень. Файли реалізуються як ділянки на магнітних та інших носіях інформації.

Файл – це впорядкована послідовність записів, які зберігаються на носіях інформації, та має певне ім'я. Ім'я файлу складається з двох частин: власне імені та розширення (типу). Ім'я може мати до восьми символів, розширення до трьох символів. Ім'я та розширення розділяються між собою крапкою.

Розширення файлу використовується для класифікації файлів та визначення належності файлу до деякої групи із загальними спільними властивостями.

Визначення типу здійснює або користувач, або програма, що породжує файл. Ім'я та тип файлу можуть містити латинські букви, а також символи: #,\$,%,^,&,(,),-_,@,!,",~. Символи з кодами менше, ніж 32, а також символи * та ? не можуть використовуватися в іменах файлів.

Справа в тому, що символи * та ? використовуються у шаблонах. *Шаблон імені файлу* часто вживається в командах для позначення відразу декількох файлів (там, де це припустимо і має сенс) або для скорочення запису імен файлів. Знак * означає, що, починаючи з даної позиції до кінця імені або розширення, можуть розташовуватися будь-які символи, що допускаються в іменах і типах файлів. Знак питання дозволяє "замаскувати" будь-який символ в одній позиції імені або типі файлу. Наприклад,

- AB*.* – усі файли з іменами, що починаються з AB;
- *.COM – усі файли типу COM;
- *.* – усі файли поточного каталогу;
- ?????.TXT – усі текстові файли із п'ятибуквеними іменами.

Каталог файлів (директорій) містить інформацію про групу файлів, збережених спільно на одному носії. Каталог має ім'я і, у свою чергу, може бути розміщений в іншому каталозі. У цьому випадку він стає підлеглим або підкаталогом. Так утворюється ієрархічна (деревоподібна) файлова система. На кожному дисковому носії завжди є кореневий каталог, що створюється при форматуванні диска й у який починають реєструватися звичайні файли і підкаталоги 1-го рівня. У них, у свою чергу, можуть реєструватися звичайні файли і підкаталоги 2-го рівня і т.д. Ланцюжки взаємо-підпорядкованих каталогів позначаються їх іменами, що розділяються знаком \. Якщо цей знак стоїть перед першим ім'ям, значить "вище" по ієрархії знаходиться тільки кореневий каталог. Наприклад,

- \WORK – WORK знаходиться на 1-му рівні;
- \WORK\PRO – PRO знаходиться на 2-му рівні.

Диски (зовнішні накопичувачі) іменуються латинськими буквами A, B, C і т.д. При зверненні до операційної системи до цих букв додається двокрапка, так що фактично логічні імена накопичувачів мають вигляд - A:, B:. Імена накопичувачів звичайно розподіляються таким чином:

- A: і B: вказують на НГМД;
 - C: і т.д. вказують на НЖМД, CD-ROM та віртуальні диски.
- Основні модулі MS-DOS.*

Властивість модульності MS-DOS дозволяє ізолювати одну від одної окремі частини великої й складної програми, полегшуючи її розробку, а також дозволяє об'єднати в кожному модулі певні логічно зв'язні групи функцій.

MS-DOS складається з наступних модулів:

- базова система введення-виведення (BIOS);
- блок початкового завантаження (Boot Record);
- модуль розширення базової системи введення-виведення (IO.SYS);
- модуль опрацювання переривань (MS-DOS.SYS);
- командний процесор (command.com);
- зовнішні команди (format.com, diskcopy.com та інші);
- драйвери зовнішніх пристроїв (ANSI, RAMDRIVE та інші).

Базова система введення-виведення.

Базова система введення-виведення розміщується в постійній пам'яті комп'ютера. Цей модуль переривань є одночасно і частиною апаратури й частиною операційної системи.

Основні функції BIOS:

1. Автоматичне тестування основних компонентів ПК при його ввімкненні.
2. Виклик блоку початкового завантаження DOS з системного диска до оперативної пам'яті комп'ютера.
3. Обслуговування переривань, які потребують безпосереднього управління апаратними компонентами.

Блок початкового завантаження.

Блок початкового завантаження – це програма, що завантажує два модулі операційної системи:

- модуль розширення базової системи введення-виведення (IO.SYS);
- модуль опрацювання переривань (MS-DOS.SYS).

Boot Record завжди розміщується на нульовій доріжці першого сектора системного чи жорсткого диска.

Модуль розширення базової системи введення-виведення (IO.SYS).

Файл IO.SYS є доповненням до базової системи введення-виведення і завантажується до оперативного запам'ятовуючого пристрою завантажувачем операційної системи та залишається в пам'яті комп'ютера постійно. Він дозволяє під'єднувати до BIOS драйвери, що обслуговують допоміжні зовнішні пристрої. Про необхідність приєднання говориться у файлі конфігурації CONFIG.SYS³, який опрацьовується файлом IO.SYS.

Модуль опрацювання переривань (MSDOS.SYS або IBMDOS.COM).

Цей модуль є основним модулем операційної системи. Він забезпечує взаємодію з програмами користувача. Файл MSDOS.SYS завантажується до ОЗП комп'ютера завантажувачем і залишається там постійно.

Компонентами даного модуля є програми, які забезпечують роботу файлової системи, інформаційний обмін з дисками, а також обслуговування деяких спеціальних ситуацій.

Командний процесор (COMMAND.COM).

Командний процесор зберігається у вигляді файлу з іменем command.com зазвичай на тому диску, з якого завантажується операційна система. Командний процесор виконує наступні функції:

1. Прийняття та синтаксичний розбір вказівок, що вводяться з клавіатури або командного файлу.
2. Виконання внутрішніх вказівок DOS.
3. Завантаження і виконання зовнішніх вказівок DOS і прикладних програм користувача, що мають розширення .com, .exe, .bat.

Внутрішні команди виконуються безпосередньо самим процесором. Для розпізнавання команди, вона внесена в таблицю імен команд. Якщо команда відсутня, то вона трактується як зовнішня. В цьому випадку командний процесор організує пошук файлу, ім'я якого співпадає з іменем команди, і запускає його для виконання.

Командний процесор складається із трьох частин.

Перші дві частини розміщуються в пам'яті слідом за файлом MSDOS.SYS і є резидентними. Перша виконує ті ж функції, що й MSDOS.SYS. Друга частина має єдину функцію – запуск файлу AUTOEXEC.BAT⁴.

Третя частина процесора транзитна. Вона включає в себе інтерпретатор команд – програми, що реалізують внутрішні команди DOS.

Коли виникає необхідність в інтерпретаторі команд, то резидентна частина перевіряє, чи знаходиться інтерпретатор команд в пам'яті і якщо він відсутній, то вона відновлює нерезидентну частину.

Зовнішні вказівки.

Зовнішні команди DOS реалізуються програмами, що зберігаються на диску у вигляді окремих завантажуваних файлів. Для виконання зовнішніх вказівок командний процесор шукає на дисках програму з відповідним іменем і, якщо знаходить її, то завантажує до пам'яті і передає їй керування. Після закінчення роботи програми командний процесор вилучає програму з пам'яті та виводить повідомлення про готовність до виконання наступних вказівок.

³ За допомогою файла конфігурації можна розширювати операційну систему і змінювати деякі параметри, що впливають на роботу зовнішніх пристроїв. Одна з найважливіших властивостей DOS полягає у можливості додавання нових зовнішніх пристроїв і підключення спеціальних програм, що забезпечують керування їхньою роботою. Таким чином, за допомогою файла конфігурації користувач може задати власні, нестандартні можливості, які діють протягом поточного сеансу роботи.

⁴ Файл AUTOEXEC.BAT є звичайним пакетним файлом. Проте він грає особливу роль, тому що при початковому запуску та ініціалізації системи, слідом за опрацюванням CONFIG.SYS, операційна система шукає в кореневому каталозі системного диска файл AUTOEXEC.BAT і автоматично його опрацьовує. Це дозволяє заносити до нього різноманітні команди, які здійснять необхідне налаштування системи.

Драйвери зовнішніх пристроїв.

Драйвери пристроїв – це спеціальні програми, які доповнюють систему введення-виведення і забезпечують обслуговування допоміжних пристроїв або нестандартне використання пристроїв. Драйвери зберігаються у вигляді файлів, імена яких мають розширення SYS. Вони завантажуються до пам'яті комп'ютера при завантаженні операційної системи, а їх імена вказують в спеціальному файлі config.sys.

Порядок завантаження MS-DOS.

Процес завантаження ОС можна умовно розділити на такі етапи.

1. Активізується програма BIOS, яка перевіряє обладнання. Якщо виявляється помилка, то виводиться відповідне повідомлення. Якщо помилка критична, то процес завантаження припиняється.

2. Після закінчення тестування програма BIOS завантажує до ОЗП програму-завантажувач (BOOT RECORD) і передає їй управління.

3. Завантажувач переглядає каталог системного диску, знаходить файли IO.SYS та MSDOS.SYS і завантажує до ОЗП ці модулі.

4. Управління передається модулю IO.SYS, який шукає файл конфігурації системи config.sys, читає його до пам'яті та аналізує. У відповідності з вказівками, які містяться в цьому файлі, завантажуються драйвери пристроїв і встановлюються параметри операційної системи. Якщо файл config.sys відсутній, то встановлюється стандартна конфігурація операційної системи.

5. Модуль IO.SYS передає управління модулю MSDOS.SYS, який встановлює таблиці переривань і проводить підготовку до завантаження командного процесора.

6. Модуль IO.SYS завантажує файл командного процесора command.com і передає йому управління. Файл command.com виконує командний файл autoexec.bat, в якому містяться вказівки і програми, які виконуються при кожному запускові комп'ютера.

Після виконання файлу autoexec.bat процес завантаження MS-DOS закінчується, і на екрані висвітлюється запрошення, яке вказує, що операційна система готова до прийому вказівок.

Приклади основних команд ms-dos 6.22.

Після завантаження операційної системи на екрані з'являється редактор командного рядка, наприклад, C:\>, в якому вводяться команди операційної системи.

Задача	Дія
Перейти на диск A:	A:
Прочитати каталог на диску	Dir
Прочитати каталог ASD	Dir ASD
Вивести інформацію на екран порціями	DIR/p
Вивести інформацію на екран в п'ять колонок в скороченій формі	DIR/W
Вивести вміст кореневого каталогу, не виходячи із поточного каталогу	DIR\
Перейти на один каталог угору	cd..
Перейти на самий верхній рівень	Cd\
Перейти у каталог XOM	Cd XOM
Відформатувати дискету	format a:
Підготувати системну дискету	format a:/s
Скопіювати одну дискету a: на c:	diskcopy a: c:
Скопіювати файл text.txt на диск a:	copy text.txt a:\
Скопіювати каталог XOM в COX	copy \ XOM\ COX
Вивести текстовий файл на друк TEXT.TXT	copy text.txt prn:
Знищити усі файли каталогу XOM	del XOM
Знищити файл tt.txt з підтвердженням	del tt.txt/p
Вивести на екран вміст текстового файлу text.txt	type text.txt
Перейменування каталогу XOM на ON	ren XOM ON
Перейменування файлів text.txt на tx.txt	ren text.txt tx.txt
Друкування файлу text.txt	print text.txt
Запуск програми tetris.exe на виконання	tetris.exe
В активному каталозі створити підкаталог P	MD P

В кореновому каталозі активного диска створити підкаталог 1	MD/1
3 активного підкаталогу вилучити підкаталог 1	RD 1
Вилучити підкаталог P2 із підкаталогу NC пристрою D:	RD D:\NC\P2
Очистити екран дисплею	CLS
Вивести каталог файлів на екран посторінково. Виведення кожної наступної сторінки ініціюється натисканням будь-якої клавіші	DIR:MORE
Вивести інформацію про атрибути файлу FILENAME з активного каталогу	ATTRIB FILENAME

2 Теоретичні відомості з загальної методики навчання інформатики, що використовуються в лабораторній роботі

Диференціація та індивідуалізація навчання в школі

Під **диференціацією** розуміють систему навчання, при якій кожен учень, оволодіваючи деяким мінімумом загальноосвітньої підготовки, що є загальнозначимою і що забезпечує можливість адаптації в життєвих умовах, що постійно змінюються, отримує право й гарантовану можливість приділяти переважну увагу тим напрямам, які в найбільшій мірі відповідають його нахилам.

Існують два види диференціації.

Перший полягає в тому, що навчаючись в одному класі за однією програмою, учні можуть засвоювати матеріал на різних рівнях. Цей вид отримав назву **рівневої диференціації**.

Другий вид – це диференціація за змістом. Вона передбачає навчання різних груп учнів за програмами, що відрізняються глибиною викладу матеріалу, об'ємом відомостей і переліком питань. Цей вид називають **профільною диференціацією**.

В основній школі переважає рівнева диференціація. В старших класах пріоритет віддають профільному вивченню предметів.

Основними цілями **індивідуалізації** навчання є:

- 1) розвиток і використання в навчанні індивідуальних якостей учня;
- 2) розвиток і використання в навчанні пізнавальних інтересів учня;
- 3) розвиток і використання в навчанні інтелектуальних здібностей і талантів кожного учня;
- 4) оптимальний розвиток здібностей до навчання у кожного учня;
- 5) підготовка до свідомого вибору професії;
- 6) розвиток у учня навичок самостійної навчальної діяльності.

Існують три основних напрями для індивідуалізації навчання:

- 1) створення відносно однорідних за складом класів чи груп учнів;
- 2) проходження курсу інформатики в прискореному чи уповільненому темпі;
- 3) класна індивідуалізація навчальних завдань в звичайному різномірному класі.

3 Завдання до лабораторної роботи

1. Повторити теоретичний матеріал з курсу інформатики по операційній системі MS DOS.
2. Опрацювати відповідний матеріал шкільного посібника.
3. Завантажити тестуючу систему та виконати тести з теми MS DOS.
4. Виконати в групі (2-3 студенти) одне з наступних завдань. Після виконання провести колективне обговорення результатів.
 - 1) Вибрати типи уроків та розробити їх структуру. Підібрати відповідні методи, засоби та прийоми навчання.
 - 2) Розробити оптимальний план викладу нового матеріалу теми.
 - 3) Вияснити, як і яку наочність можна використати при вивченні теми? Які проблемні ситуації можуть бути організовані та які шляхи їх розв'язання?
 - 4) Які форми можуть бути використані для закріплення вивченого матеріалу? Який навчальний матеріал буде для цього використаний?
 - 5) Які записи будуть пропонуватися для учнівських зошитів? Який матеріал буде подаватися на класній дошці? Що буде опрацьовуватися з допомогою комп'ютера?
5. Оформити звіт по лабораторній роботі.

Методика вивчення операційної системи MS Windows

ОСНОВНА МЕТА	Розгляд основних методичних особливостей вивчення операційної системи MS Windows в школі.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ	Знати основні принципи, форми, прийоми та засоби вивчення операційної системи MS Windows.
ППЗ	Операційна система Windows. Тестуюча система IQ Тест.

1. Методичні рекомендації та теоретичні відомості

Загальні відомості про операційну систему Windows

Операційна система Windows фірми Microsoft являє собою графічне псевдомультимедійне середовище. Це сучасна операційна система, яка забезпечує можливість одночасного виконання декількох завдань та плавного перемикання між ними. Коли йдеться про псевдобагатозадачність, то мається на увазі, що насправді програми виконуються у системі по черзі (сам процесор перемикається по черзі між ними), але за рахунок дуже швидкої зміни такої черги, моделюється режим одночасної роботи.

Система має зручний графічний інтерфейс. Для цього використовується аналогія з *робочим столом*.

Всі елементи, з якими працює користувач є об'єктами. До об'єктів відносяться програми, файли, папки диски, значки, ярлики тощо.

Робота в ОС Windows передбачає наявність на робочому місці маніпулятора "миша". Більшість дій можна також виконати і за допомогою клавіатури, однак застосування миші простіше і ефективніше.

Щоб виділити об'єкт на екрані, необхідно перемістити на цей об'єкт вказівку миші (біла стрілка), а потім натиснути та відпустити ліву кнопку миші. Щоб перемістити об'єкт на екрані, необхідно перемістити на об'єкт вказівку миші, натиснути та, утримуючи в такому положенні ліву кнопку миші, перемістити об'єкт на нове місце і відпустити кнопку миші.

Натискання лівою кнопкою миші на будь-якому об'єкті робочого стола робить цей об'єкт поточним (активним).

Подвійне натискання лівою кнопкою миші на значку:

- папки – розкриває відповідне їй вікно;
- документа – запускає на виконання прикладну програму, за допомогою якої був створений цей документ, та завантажує його зміст;
- прикладної програми – запускає її на виконання (завантажує).


Натискання правою кнопкою миші на будь-якому об'єкті виводить на екран контекстне меню. Контекстне меню містить у собі перелік певних дій, які можуть бути виконані, а також пункт *Свойства*. Обравши цей пункт, користувач може переглянути і змінити характеристики поточного об'єкта. Контекстне меню залежно від ситуації може змінюватися. Це пов'язано з тим, що з різними об'єктами в різні моменти часу можна виконувати різні операції та дії.



Головна ідея роботи у системі Windows полягає в наступному:


- всі операції виконуються з об'єктами, які у відкритому стані являють собою вікна;
- всі види робіт виконуються у вікнах;
- правила роботи з усіма вікнами подібні.

Заголовок вікна містить назву програми.

Кнопки управління вікном дозволяють:

 – згорнути вікно (після цього вікно матиме вигляд кнопки на *панелі задач*, натиснувши по якій маніпулятором миша, можна відновити положення та розмір вікна; робота додатка не припиняється);

 – розгорнути вікно на весь екран (після цього кнопка набуде іншого вигляду , що дозволяє відновити початкове положення та вигляд вікна);

 – закрити вікно і припинити роботу додатка.

У рядку *Меню* обов'язковими є пункти *Файл* (для роботи з дисками, файлами тощо), *Правка* (для роботи з об'єктами), *Вид* (параметри перегляду інформації), ? (довідкова інформація).

Розмір та положення вікна на робочому столі можна змінювати за допомогою маніпулятора миша.

Якщо вікно документа є розгорнутим, то воно співпадає з вікном програми. У цьому випадку існує два набори кнопок керування розміром вікна: верхні призначені для управління вікном самої програми, а нижні – вікном відкритого документу.

Розглянемо деякі об'єкти ОС Windows.

Файл - логічно об'єднана інформація (програма або організована сукупність даних), яка займає деяку область пам'яті дискового простору і має певне ім'я. *Ім'я* файлу складається з двох компонентів, розділених крапкою: власне імені (від 1 до 255 символів) та розширення імені (зазвичай - від 0 до 3 символів). *Розширення* імені файлу визначає сам користувач або програма, за допомогою якої було створено файл. Основне ім'я може складатися з літер, цифр, пропусків, а також символів ~ ! @ # \$ % & () _ = - + [] { } ; , ' (порівняйте з ОС DOS!).

У файлової системи файли можна об'єднувати в групи, надаючи такій групі певне ім'я. Така група називається папкою (аналог директорія у ОС DOS).

Папка – це область на носії, що містить інформацію про файли, об'єднані в групу за деякою ознакою. Папка має конкретне ім'я і може зберігатися у іншій папці разом з усіма файлами. Так створюється ієрархічна файлова структура, яку іноді називають деревоподібною. Ця структура подібна до структури звичайного бібліотечного каталогу.

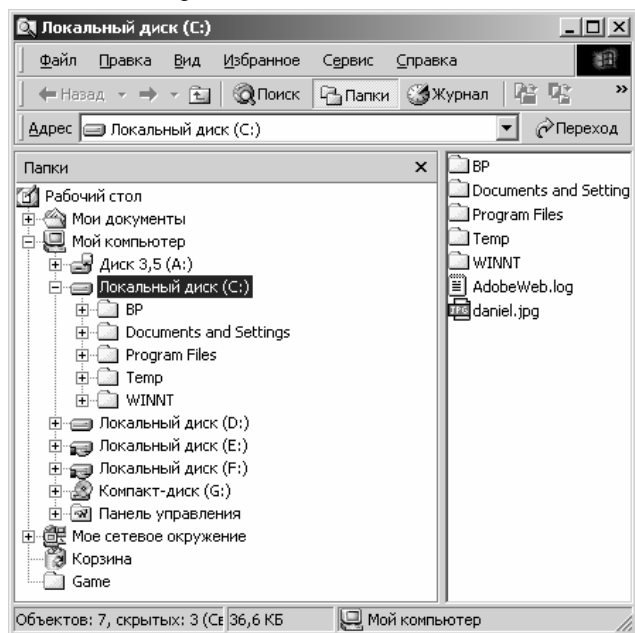
Документ – це будь-який файл, який вміщує дані: текст, графічне зображення, електронну таблицю тощо.

Ярлик – це посилання на деякий об'єкт, який розташований у певному місці файлової системи. Ярлики використовують для запуску програм або швидкого доступу до файлів та папок. Ярлик не є об'єктом у повному розумінні. Ярлик – це лише посилання на деякий об'єкт.

Значок – використовується для подання об'єктів і визначення їх типу. Значок ярлика відрізняється тим, що у лівому куті значка є маленький квадрат, усередині, якого розміщена стрілка – ознака того, що це ярлик.

Основними типами об'єктів, з якими працює користувач в ОС Windows, є папки, документи (файли), ярлики. При роботі з об'єктами їх часто доводиться копіювати, видаляти, переміщувати, перейменовувати, відшукувати у файлової системі. Для цього часто використовують програму *Проводник* (хоча є й інші програми для роботи з об'єктами, наприклад, FAR, Total Commander тощо). Майже всі дії можуть бути виконані як з окремими об'єктами, так і з їх групою. Працюючи з програмою *Проводник*, найзручніше використовувати маніпулятор миша: ліву кнопку – для зазначення об'єкта, запуску програми або підтвердження своїх дій, і праву – для використання контекстного меню.

Ліва панель вікна програми *Проводник* відображає деревоподібну структуру дисків і папок ПК. Одна з папок є активною (відкритою). На мал.8 – диск C:.



Мал. 8.

документи, а також робочий стіл. Найзручніше для створення об'єктів використовувати контекстне меню: обравши у ньому пункт *Создать*, отримаємо додаткове меню із переліком об'єктів, які можна створити.

Документи, як правило, створюються у відповідних прикладних програмах (без використання контекстного меню): для створення текстового документу необхідно запустити програму текстового редактора; для створення малюнків – програму графічного редактора тощо. Для створення документів можна також використати контекстне меню. Обравши у ньому пункт *Создать*, обираємо у додатковому меню тип створюваного документу (текстовий документ, документ Word, лист Microsoft Excel тощо).

Для збереження в одному місці деякої кількості однотипних документів (файлів) зручно використовувати папку. **Папку** створюють за допомогою контекстного меню. Якщо при виконанні команди *Создать* у контекстному меню обрати рядок *Папка*, то у вікні чи на робочому столі з'явиться піктограма, біля якої розміщене затемнене поле. У це поле потрібно ввести ім'я нової папки.

Ярлики створюються аналогічно до папок. Для того, щоб створити на робочому столі ярлик необхідно у вільному від об'єктів місці робочого стола натиснути праву клавішу миші і у контекстному меню виконати послідовність команд *Создать* → *Ярлык*. У вікні, що відкриється (мал. 9), потрібно скористатися кнопкою вікна *Обзор*. При натисканні цієї кнопки відкривається діалогове вікно *Обзор* (мал. 10), що дозволить швидко знайти шлях до об'єкта. Після натискання кнопки вікна *Открыть* відбувається повернення до вікна *Создание ярлыка*, а інформацію про розташування об'єкта буде відображено у рядку *Командная строка* цього вікна. Для продовження роботи використовують кнопку вікна *Далее*.

Якщо біля папки є значок +, то це означає, що папка містить у собі вкладені папки, які не відображені на лівій панелі (на мал.10 – диск D:).

Якщо біля папки є значок -, то вкладені папки є відображеними (на мал.8 – диск C:).

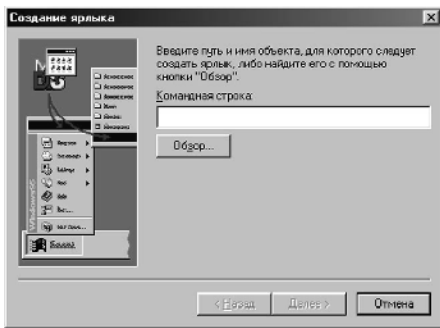
Якщо біля папки немає таких значків, то це означає, що ця папка не містить жодної вкладеної папки (але вона може містити вкладені файли)

Для того, щоб відобразити вкладені папки потрібно натиснути лівою кнопкою миші по значку +. Для того, щоб приховати вкладені папки потрібно натиснути лівою кнопкою миші по значку -.

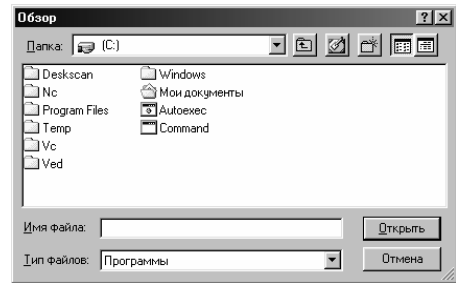
Для того щоб відкрити папку (переглянути її вміст на правій панелі), потрібно зробити її активною, натиснувши на її зображенні лівою кнопкою миші.

Права панель вікна відображає вміст відкритої папки, де, крім вкладених папок, міститься також перелік файлів відкритої папки.

Для створення об'єктів найчастіше використовують вікно програми *Проводник*, папки *Мой компьютер*, *Мои*



Мал. 9.



Мал. 10.

Виокремити групу об'єктів у програмі *Проводник* можна таким чином:


- якщо об'єкти розташовані підряд, потрібно: один раз натиснути лівою кнопкою миші на значку першого з об'єктів і, утримуючи клавішу Shift на клавіатурі, натиснути лівою кнопкою миші на значку останнього з об'єктів ;
- якщо об'єкти розташовані окремо, потрібно: утримуючи клавішу Ctrl на клавіатурі, за допомогою лівої кнопки миші зазначити об'єкти, що включаються до групи.

Аналогічно можна виокремити групу об'єктів на робочому столі або у будь-якій із відкритих папок.


Для того, щоб *перейменувати об'єкт*, необхідно натиснути по ньому правою клавішею мишки, у контекстному меню обрати пункт *Переименовать* і внести у назві об'єкта потрібні зміни.

Копіювати об'єкти у програмі *Проводник* можна декількома способами. Найбільш уживаними є копіювання спосіб перетягування та за допомогою контекстного меню.

Копіювання способом перетягування:


- у правій панелі програми виокремити потрібний об'єкт (або їх групу);
- натиснути праву кнопку миші (вказівка миші повинна мати вигляд  і знаходитися на області виокремлених об'єктів) і не відпускаючи її почати перетягування в ліву область вікна, поки активною (виділеною синім кольором) не стане папка, в яку потрібно скопіювати об'єкт (або їх групу);
- відпустити праву кнопку миші і у контекстному меню, що з'явиться, обрати пункт *Копировать*.

Копіювання за допомогою контекстного меню:

- у правій панелі програми виокремити потрібний об'єкт (або їх групу);
- клацнути правою кнопкою миші (вказівка миші повинна мати вигляд  і знаходитися на області виокремлених об'єктів) і у контекстному меню, що з'явиться, обрати пункт *Копировать*;
- відкрити папку, у яку потрібно вставити скопійований об'єкт (або їх групу);
- у вільному від об'єктів місці натиснути праву кнопку миші і у контекстному меню, що з'явиться, обрати пункт *Вставить*.

Переміщення об'єктів подібне до копіювання, тільки замість функції *Копировать* слід вибрати пункт *Переместить*.

Знищити об'єкт (або їх групу) можна також декількома способами:

- виокремити поодинокий об'єкт (або їх групу) і натиснути клавішу *Delete* на клавіатурі;
- виокремити поодинокий об'єкт (або їх групу) та обрати у контекстному меню пункт *Удалить*;
- виокремити поодинокий об'єкт (або їх групу), натиснути ліву кнопку миші (вказівка миші повинна мати вигляд  і знаходитися на області виокремлених об'єктів) і не відпускаючи її почати перетягування у папку *Корзина* на робочому столі.

У разі виконання будь-якого із цих способів об'єкт буде переміщено у спеціальну папку на робочому столі – *Корзина*. За потребою можна відновити знищений об'єкт. Для цього необхідно:

- відкрити папку *Корзина*;
- виокремити поодинокий об'єкт (або їх групу) та у меню виконати команду *Файл → Восстановить*.

Якщо очистити *Корзину* (виконати у меню команду *Файл → Очистить корзину*), то відновлення стає неможливим.

2. Завдання до лабораторної роботи

1. Повторити теоретичний матеріал з курсу інформатики по операційній системі Windows.
2. Опрацювати відповідний матеріал шкільного посібника.
3. Завантажити тестуючу систему IQ Тест та виконати тести з теми MS Windows.
4. Здійснити **логіко-дидактичний аналіз теми** шкільного посібника [5] за *схемою*:

- 1) визначити мету вивчення теми;

Мету слід розуміти як передбачення результатів і тих дій, які ведуть до досягнення цих результатів. Під результатами розуміються не тільки змістовні факти, але й уміння виконувати дії, що ведуть до усвідомленого та глибокого оволодіння фактами в різноманітній навчальній діяльності.

- 2) сформулювати основні знання та вміння, якими повинні оволодіти учні (див. Програму шкільного курсу інформатики);
- 3) скласти фрагмент тематично-календарного плану вивчення теми, врахувавши самостійні та контрольні роботи, уроки розв'язування задач, лабораторні роботи, заліки, інші форми роботи.
- 4) логічний та фактичний аналіз змісту теми (теоретичного, практичного матеріалу та змісту задач теми):

Виділити всі основні поняття, означення та вміння з коротким їх описом або поясненням.

Проаналізувати повноту, логічність, послідовність та відповідність матеріалу шкільного посібника вимогам програми шкільного курсу інформатики.

Проаналізувати задачі до вивчення теми:

- *яка кількість задач сприяє розкриттю, конкретизації, поглибленню фактичного матеріалу теми?*
- *як згруповані задачі у відповідності до матеріалу теми?*
- *чи можна назвати декілька задач, на розв'язок яких повинна бути сконцентована увага учнів з наступним закріпленням прийомів і методів розв'язку таких задач?*
- *чи є задачі на здійснення пошуку розв'язку? Скільки таких задач? Чи достатньо їх для досягнення поставлених цілей?*
- *чи є задачі з нестандартною фабулою, проблемні, цікаві?*

- 5) сформулювати основні навчальні задачі і відібрати відповідні навчально-пізнавальні дії (наприклад, розкрити структуру та призначення основних елементів робочого столу Windows, навчально-пізнавальні дії – робота з підручником, конспектом, таблицями, комп'ютером);
 - 6) відбір основних засобів (комп'ютер, таблиці тощо), методів (робота з підручником та додатковою літературою, пояснення матеріалу, самостійне оволодіння матеріалом) і прийомів навчання (засвоєння теоретичних основ, практична робота, лабораторна робота тощо);
 - 7) визначити форми контролю і оцінки процесу та результатів навчальної діяльності учнів (що буде контролюватися і як буде контролюватися).
5. Оформити звіт по лабораторній роботі.

ЛАБОРАТОРНА РОБОТА №8 (4 год.)

Методика вивчення основ роботи з дисками

ОСНОВНА МЕТА	Розгляд основних методичних особливостей вивчення роботи з дисками: форматування, архівація даних, антивірусні програми.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ	Знати основні принципи, форми, прийоми та засоби вивчення роботи з дисками в школі.
ППЗ	Архіватори WINRAR та WINZIP. Файл FORMAT.COM

1. Методичні рекомендації

Інформація на магнітних дисках зберігається у вигляді файлів. Файл – це поійменована область пам'яті на диску, в якій зберігаються однотипні дані під одним ім'ям.

Перед використанням диски формують, тобто розбивають їх на доріжки та сектори. Мінімальною порцією запису даних на диск та читання даних із диска є один *кластер*. Наприклад, на дисках об'ємом в 720 Кб 1 кластер = 2 секторам, а на дискетах об'ємом 1,44 Мб 1 кластер = 1 сектору, на вінчестерах 1 кластер може дорівнювати 4, 8, 16, 32 секторам. Кожний файл на диску займає цілу кількість кластерів. Зрозуміло, що розмір файлу не буде пропорційним одному кластеру, а це означає, що останній кластер кожного файлу є, як правило, незаповненим.

Форматування дисків проводиться командою **FORMAT** з середовища MS DOS або командою **Форматувати** контекстного меню у Windows. Якщо жорсткий диск має декілька розділів (C, D, E...), то для форматування розділу C у MS DOS завжди користуються ключем \S. Інші диски формуються без нього. Це пов'язано з тим, що диск C, як правило, є системним. Випадкове форматування жорсткого диска небезпечно, оскільки можна втратити всю записану на ньому інформацію. Однак для таких випадків операційна система передбачила два ступені захисту. Перший - в разі спроби формувати жорсткий диск, на якому є інформація, запитується ім'я тому. Другий - перед початком форматування виводиться повідомлення про те, що всі дані диску будуть знищені.

На кожному диску є системна область та область даних. Системна область складається з нульового сектора, таблиці розташування файлів FAT (File Allocation Table), яка в силу її важливості зберігається в двох копіях, області кореневого каталогу.

З метою економії місця при збереженні документів, збереження резервних копій програм, утримання резервних копій на інших носіях на випадок попередження втрати інформації при виході з ладу ПК використовують спеціальні програми – **архіватори**.

Зберігання архівів у початковому вигляді дуже не вигідно через великі обсяги інформації. Річ у тому, що при зберіганні інформації в початковому вигляді неефективно використовується поверхня носія. Доцільно вихідну інформацію попередньо стиснути, а потім вже створювати її копії. *Такий процес створення архівних копій називається архівацією*. Завдяки архівації досягається економія дискового простору від 20 % до 90%, що дозволяє на одному і тому самому носії зберігати значно більший обсяг інформації.

Більшість із цих програм не потрібно спеціально купувати, так як вони пропонуються із програм загального користування (Shareware) або вільного поширення (Freeware).

Принцип роботи архіватора ґрунтується на здійсненні пошуку у файлі “надлишкової” інформації і наступному її кодуванню з метою отримання мінімального об'єму. Найбільш відомий метод архівації файлів є стискання послідовностей однакових символів. Наприклад, всередині вашого файлу знаходиться послідовність байтів, котрі часто повторюються. Тому, щоб не зберігати кожен байт, фіксують кількість символів, що повторюються і їх позиції.

Нехай файл містить 15 байт і має наступні символи B B B B L L L L A A A A.

В шістнадцятковій системі 42 42 42 42 42 4C 4C 4C 4C 4C 41 41 41 41

Архіватор може представити цей файл в наступному вигляді (в шістнадцятковій системі числення) 01 05 42 06 4C 0A 05 41

Цю послідовність можна інтерпретувати наступним чином: з першої позиції п'ять разів повторюється символ “B”, з позиції 6 повторюються символ “L” і з позиції 11 п'ять разів повторюється символ “A”. Отже, для зберігання файлу в новій формі потребується 9 байт, що на 6 байт менше початкового значення.

Такий метод не забезпечує великої економії місця, якщо невелика кількість символів, що повторюються. Більш практичний метод стикування даних – це, так званий оптимальний префіксний код, і зокрема, кодування символами змінної довжини (алгоритм Хафмана). Код змінної довжини дозволяє записувати символи і групи символів, котрі найбільше трапляються лише кількома бітами, в той час як символи, що зустрічаються рідше, будуть записані більш довгими бітовими рядками. Наприклад, аналізуючи довільний англійський текст, ми помітимо, що E зустрічається частіше ніж Z. Таким чином, використавши спеціальну таблицю відповідності, можна закодувати кожен літер E меншим числом бітів і використати більш довгий код для літер, що зустрічаються рідше.

Архіватори ARJ, PAK, LHARC, PKZIP працюють на основі алгоритма Лемпела-Зіва. Ці архіватори класифікуються як адаптивні словникові кодувальники, в котрих текстові рядки замінюються вказівниками на ідентичні до них рядки, що зустрічалися раніше в тексті. Наприклад, всі слова цієї книги можна представити у вигляді номерів сторінок і номерів рядків

деякого словника. Особливою рисою цього алгоритму є використання граматичного аналізу попереднього тексту з розкладанням його на фрази, котрі записуються в словник.

Вказівки дозволяють робити посилання на довільну фразу у вікні визначеного розміру, яка передувала поточній фразі. Якщо відповідність буде знайдено, то поточна фраза замінюється вказівником на свого попереднього двійника. Ступінь стискання файлів дуже залежить від формату файлу. Деякі графічні файли вже при їх створенні стискаються тому подальше стискання виграшу в розмірах нічого не дає.

Перші архіватори з'явилися в 1985 році.

Функціональні можливості архіваторів різноманітні, але є ряд загальних властивостей, зокрема:

- *можливість додавання файлів в архів,*
- *поновлення архіву,*
- *перегляд файлів в архіві,*
- *знищення файлів в архіві,*
- *захист файлів від несанкціонованого доступу,*
- *вилучення файлів з архіву,*
- *перевірка цілісності архіву.*

Архівний файл може використовуватися лише після того, як він буде відновлений у початковому вигляді, тобто розархівований.

Для поновлення архівів використовують ці ж самі програми, або спеціальні програми відновлення з архіву.

Найвчені архіватори під DOS та Windows. Найбільш популярні RAR та ZIP.

2. Завдання до лабораторної роботи

1. Повторити теоретичний матеріал по роботі з дисками з курсу інформатики.
2. Опрацювати п. 4.4, 4.5 §4 глави 3 та п. 2.10 §2 глави 4 шкільного посібника [5].
3. Розробити **інструкцію** до шкільної **лабораторної роботи** з теми "Форматування диска. Архівація даних" за **схемою**:
 - 1) *Тема роботи.*
 - 2) *Мета роботи.*
 - 3) *Обладнання та програмне забезпечення.*
 - 4) *Короткі теоретичні відомості.*
 - 5) *Завдання до лабораторної роботи та план роботи.*
 - 6) *Форма представлення результатів.*
 - 7) *Контрольні запитання.*
 - 8) *Вимоги до оцінювання результатів роботи.*
4. Виконати розроблену лабораторну роботу (звіт представити).
5. За результатами при потребі зробити корективи в інструкції.
6. Дати відповідь на контрольні питання.
7. Оформити звіт по лабораторній роботі.

Контрольні запитання

1. В чому полягає процес підготовки диска до роботи?
2. Які дисководи має комп'ютер стандартної комплектації? Який об'єм мають дискети? Який об'єм мають вінчестери?
3. Якими літерами позначаються дисководи для дискет? Якими літерами позначаються логічні диски на вінчестері (або кілька вінчестерів)?
4. Як взнати, скільки вільного місця залишилося на диску?
5. Як називається процес розподілу поверхні диску на концентричні доріжки і сектори для запису на них файлів?
6. Які початкові дії необхідно виконати при форматуванні дискети?
7. Що станеться з файлами на дискеті після її форматування?
8. Як називається одиниця обміну інформації на дисках?
9. В чому полягає операція дефрагментації?
10. Які початкові команди необхідно зробити при дефрагментації?
11. Що станеться з файлами на вінчестері або дискеті після дефрагментації?
12. Що зробити, щоб не втратити важливу інформацію при виході із ладу комп'ютера?
13. Яким чином записати інформацію великого об'єму на дискету, яка за технічними даними має невеликий об'єм?
14. Яка процедура допомагає записувати на дискету відносно великі об'єми інформації і при потребі відновлювати її?
15. Як створити архівний файл для одного файла?
16. Як створити архівний файл для кількох файлів?
17. Як створити архівний файл для папки?
18. Назвіть найбільш поширені архіватори?
19. Як добути окремий файл і архіву?
20. Як добути кілька файлів із архіву?
21. Як розархівувати архівний файл папки?
22. Які розширення найчастіше мають архівні файли?
23. Що таке архівний файл?
24. Чи можна відформатувати дискету довільним чином? Чому?
25. Опишіть, які дії необхідно виконати, щоб відформатувати дискету в ОС DOS та Windows?
26. Що розуміють під архівацією файлів?

27. Для чого призначені архіватори?
28. Як створити архів в архіваторі WinRAR?
29. Дайте характеристику графічного екрана WinZIP.
30. Дайте характеристику графічного екрана WinRAR.

Методика вивчення глобальної мережі Інтернет та її можливостей

ОСНОВНА
МЕТА

Розгляд основних особливостей
вивчення глобальної мережі Інтернет та її
можливостей в школі.

ВИМОГИ ДО ЗНАНЬ І
ВМІНЬ

Знати основні характеристики
глобальної мережі Інтернет, вміти
організувати роботу з учнями по пошуку
інформації в Інтернет.

1. Теоретичні відомості

Комп'ютерна мережа — це система розподіленого опрацювання інформації, що складається як мінімум із двох комп'ютерів, які взаємодіють між собою за допомогою *засобів зв'язку*.

Засоби зв'язку мають забезпечувати надійну передачу інформації між комп'ютерами мережі. Комп'ютери, що входять до складу мережі, виконують досить широке коло функцій, основними з яких є:

- організація доступу до мережі;
- керування передачею інформації;
- надання обчислювальних ресурсів і послуг абонентам мережі.

При розгляді комп'ютерних мереж широко використовуються поняття: *клієнт* і *сервер*. На структурному рівні під *сервером* розуміється комп'ютер, що надає свої ресурси іншим комп'ютерам — *клієнтам*.

На програмному рівні під *сервером* і *клієнтом* розуміються процеси (програми), що виконують, відповідно, функції надання і використання мережних ресурсів. Крім того, комп'ютери, через які користувачі одержують доступ до ресурсів комп'ютерної мережі, прийнято називати *робочими станціями*.

Комп'ютерні мережі поділяються на глобальні і локальні. Глобальні мережі охоплюють значні території. Наприклад, мережа Internet охоплює всю земну кулю. Локальна мережа розміщується в межах окремої організації або корпорації.

Будь-яка комп'ютерна мережа характеризується своєю архітектурою, яка визначається її *топологією, протоколами, інтерфейсами, мережними технічними і програмними засобами*.

Топологія комп'ютерної мережі відображає структуру зв'язків між її основними елементами. Через низку причин існує відмінність між топологіями глобальних і локальних мереж. Топологія глобальних мереж характеризується достатньо складною, неоднорідною структурою. У свою чергу, топологія локальної мережі, зазвичай, має визначену структуру: лінійну, кільцеву або деревоподібну.

Протоколами називають правила взаємодії функціональних елементів мережі.

Інтерфейси — це засоби сполучення функціональних елементів мережі. Варто звернути увагу, що у ролі функціональних елементів можуть виступати як окремі пристрої, так і програмні модулі. Відповідно до цього, існують апаратні і програмні інтерфейси.

Під *мережними технічними засобами* розуміють різноманітні пристрої, що забезпечують об'єднання комп'ютерів в єдину комп'ютерну мережу. До цих пристроїв відносяться мережні контролери, вузли комутації та ін.

Мережні програмні засоби керують роботою комп'ютерної мережі і забезпечують відповідний інтерфейс з користувачами. До мережних програмних засобів належать мережні операційні системи та допоміжні (сервісні) програми.

Кожна із складових архітектури комп'ютерної мережі характеризує її окремі властивості, і тільки їх сукупність характеризує всю мережу загалом. Таким чином, вибір комп'ютерної мережі може бути зведений до вибору її топології, протоколів, апаратних засобів і мережного програмного забезпечення. Кожна з цих компонент є відносно незалежною. Наприклад, мережі з однаковою топологією можуть використовувати різні методи доступу, протоколи і мережне програмне забезпечення. У різних мережах можуть застосовуватись однакові протоколи і (або) мережне програмне забезпечення. Це розширює можливість вибору найбільш оптимальної архітектури комп'ютерної мережі.

Інтернет — це всесвітня комп'ютерна мережа, яка об'єднує в єдине ціле десятки тисяч різноманітних локальних та глобальних мереж.

Інтернет започаткований Агентством з наукових досліджень Міністерства оборони США в 1969 році в рамках проекту ARPA з метою розробки надійних цифрових комунікацій для військових потреб. У 90-ті роки Інтернет вийшла за межі США і до неї приєдналися країни всіх континентів світу. Інтернет — це система взаємопов'язаних мереж, до її складу входять комп'ютери найрізноманітніших типів. Для зв'язку між ними використовуються найрізноманітніші канали: телефонна лінія, супутниковий зв'язок, радіоэфір, оптико-волоконний кабель, телевізійний кабель. Взаємодія всіх об'єктів мережі забезпечується використанням спільного мережного *протоколу* — сукупності правил обміну інформацією між комп'ютерами, встановлених за взаємною угодою.

Загальноприйнятим протоколом в мережі Інтернет є TCP/IP, розроблений в 1983 році. Дата його створення вважається днем народження Internet. Фактично TCP/IP — це два протоколи: IP (Internet Protocol — міжмережний протокол) і TCP (Transmission Control Protocol — протокол управління передачею). Протокол IP розроблений для пересилання даних, упорядкованих у пакети, між мережами.

Принцип комутації пакетів був розроблений в 1968 р. і його сутність полягає в тому, що ті повідомлення, які пересилаються по мережі, впорядковуються в пакети, що містять не тільки повідомлення, але й інформацію про маршрут — адреси комп'ютера-відправника і комп'ютера-одержувача.

Спеціальні комп'ютери-маршрутизатори (router) визначають шлях, яким пакети повинні проходити від одного комп'ютера до іншого, користуючись таблицями і алгоритмами маршрутизації.

Великі повідомлення поділяються на кілька пакетів, кожний із своїм номером. За цими номерами комп'ютер-одержувач повинен повністю відновити вихідне повідомлення після одержання всіх пакетів.

Пакети передаються від одного вузла Інтернету до іншого, який розташований ближче до адресата. Пакети одного і того ж повідомлення можуть бути передані навіть різними шляхами. Якщо передача пакету була невдалою, то вона повторюється.

Протокол TCP відповідає за те, як документи розбиваються на пакети і як потім збираються докупи, а протокол IP відповідає за те, як пакети досягають адресата.

Доступ до Internet може отримати будь-який користувач комп'ютера, на якому встановлені модем та необхідне програмне забезпечення. Для підключення до Internet існують спеціальні організації — провайдери (to provide — забезпечувати) послуг Internet.

Доступ до мережі може забезпечуватися декількома способами. По-перше, додзвонювання (Dialup Connection) за допомогою звичайних телефонних ліній. Такий спосіб доступу є дешевим, але швидкість передачі даних при цьому є малою. Ефективнішим є спосіб безпосереднього додзвонювання через виділені лінії (Dedicated Line Connection). Він є дорожчим, але забезпечує набагато кращу якість зв'язку та швидкість передачі інформації.

Кожний комп'ютер, підключений до Internet, повинен мати унікальну адресу. В Internet використовуються два типи адрес: *цифрові (IP-адреси)* та *доменні (domain — область, сфера)*.

IP-адреса за змістом аналогічна поштовому індексу і є послідовністю з чотирьох чисел (октет), розділених крапками. Кожне з чисел може займати 1 байт. Ліва частина IP-адреси визначає конкретну мережу в Internet і називається ідентифікатором мережі (network ID). Права частина IP-адреси визначає конкретний комп'ютер у цій мережі й називається ідентифікатором комп'ютера (host ID).

Однак для сприйняття користувачем IP-адреса незручна. Тому користувачі звичайно працюють із доменними адресами — унікальними іменами комп'ютерів у Internet. Доменна адреса також складається з частин (доменів), розділених крапками. Доменна адреса комп'ютера включає в себе, як мінімум, два рівні доменів (кількість доменів в адресі не регламентується).

Найправіший — домен першого рівня, наступний — домен другого рівня і т.д. Домени першого рівня стандартизовані, а інші вибираються за бажанням тих, хто їх встановлює. Імена для доменів верхнього рівня видає інформаційний центр Інтернету (InterNIC), решту імен фіксують ті організації, яким такі права делеговані.

Домен першого рівня визначає країну або тип організації, до якої належить комп'ютер. Наприклад, Україна — ua; Росія — ru, освітні заклади — edu, урядові установи — gov, комерційні організації — com, провайдери послуг Internet — net тощо.

Домен другого рівня визначає організацію, яка володіє або керує мережею, що містить даний комп'ютер. У більшості випадків ім'я цього домену збігається з назвою відповідної фірми або її торговою маркою.

При роботі з електронною поштою адреси поштових скриньок користувачів формують додаванням до адреси комп'ютера, який відіграє роль поштового сервера, зліва через символ @ імені або псевдоніма користувача. Наприклад, username@zu.edu.ua.

Широкою популярністю в мережі Internet користуються такі послуги: електронна пошта (забезпечує передачу кореспонденції, що подана в електронному вигляді); групи новин UseNet (групові дискусії); служба передачі файлів між комп'ютерами (FTP-служба); IRC-служба (Internet Relay Chat) — обмін текстовими повідомленнями в режимі реального часу; WWW (World Wide Web — „всесвітня павутина”) – передача мультимедійних документів; служби Telnet, WAIS тощо.

2. Завдання до лабораторної роботи

1. Повторити теоретичний матеріал з курсу інформатики по Internet.
2. Опрацювати главу 6 шкільного посібника [5].
3. Здійснити логіко-дидактичний аналіз теми §§ 1-3 глави 6 шкільного посібника [5] (див. **схему** в лаб. роб. №7).
4. Розробити інструкцію до шкільної лабораторної роботи з теми "Пошук інформації в Інтернет" (див. **схему** в лаб. роб. №8).
5. Оформити звіт по лабораторній роботі.

ЛАБОРАТОРНА РОБОТА №10 (10 год.)

Методика вивчення текстового редактора

ОСНОВНА МЕТА	Розгляд основних методичних особливостей вивчення текстового редактора MS Word в школі.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ	Знати основні принципи, форми, прийоми та засоби вивчення текстового редактора MS Word.
ППЗ	Текстовий процесор MS Word..

Розділ 1

Набір, зберігання, редагування та форматування текстового документа

1. Теоретичні відомості

Для роботи з великим обсягом текстової інформації використовують спеціальні програми – текстові редактори.

Одним з найпоширеніших текстових редакторів є редактор *Word*, який входить до складу інтегрованого пакету *Microsoft Office*.

До основних функцій текстового редактору можна віднести:

- введення тексту та його подальше редагування;
- автоматична перевірка орфографії;
- форматування тексту (встановлення параметрів сторінки; вирівнювання тексту; визначення міжрядкового інтервалу та меж абзацу; створення і редагування колонтитулів; автоматична нумерація сторінок, розділів або пунктів; вибір певного шрифту та його характеристик тощо);
- перенесення або копіювання частини тексту;
- робота з декількома документами одночасно;
- автоматичний пошук необхідної інформації у тексті;
- автоматична заміна певного тексту за визначеним зразком;
- друкування готового тексту;
- можливість при наборі тексту вводити математичні формули, діаграми, створювати графічні зображення, виконувати вставку готових малюнків тощо.

Для запуску текстового редактора Word необхідно активізувати головне меню (натиснувши кнопку *Пуск*) і вибрати у пункті *Программы* команду *Microsoft Word*. Після запуску на робочому столі з'явиться вікно програми.

На робочому полі Word зверху знаходиться рядок заголовку, під ним рядок меню, далі панелі інструментів – *Стандартная, Форматирование* (при потребі можна встановити інші панелі), горизонтальна та вертикальна лінійки, вертикальна та горизонтальна смуги прокрутки з бігунками, внизу рядок стану. Праворуч знаходиться *Область задач*, на якій можна вибрати режими створення документа.

На екрані знаходяться два курсори: клавіатури і миші.

Курсор клавіатури знаходиться тільки в полі можливого набору тексту і служить для вказування місця набору або редагування тексту.

Курсор миші може переміщатися по всьому екрану і служить для швидкого переміщення курсору клавіатури на нове місце текстового документа, а також для подачі різноманітних команд.

Меню – це список команд, у якому можна вибрати потрібну команду для виконання.

Кожне меню має свою назву, яка відображає призначення команд, що містяться в ньому: *Файл, Правка, Вид, Вставка, Формат, Сервис, Таблица, Окно, ? (Довідка)*.

При натисканні лівою клавішею миші на назві меню, воно розгортається у список команд.

Одразу після запуску Word його вікно готове для набору тексту нового документа. При необхідності створення ще одного документа потрібно натиснути комбінацію клавіш *Ctrl + N* або виконати команди меню *Файл → Создать...* та вибрати пункт *Новый документ* на *Панели задач*, що з'явиться справа. Для переходу між вікнами різних документів використовується пункт меню *Окно*.

Word забезпечує кілька режимів роботи з текстовими документами. Вибір режиму залежить від виду створюваного документа і подальшої роботи з ним. Режими роботи з документами можуть бути *Обычный, Веб-документ, Разметка страницы, Структура, На весь экран*. Потрібний режим встановлюється із меню *Вид*.

Режим *Обычный* встановлюється автоматично при запуску програми Word. Він є найбільш придатним для введення і редагування нових документів, дає на екрані найбільше корисної площі для тексту.

У режимі *Разметка страницы*, документ подається у тому вигляді, в якому буде виглядати при друці. В цьому режимі з'являється вертикальна лінійка, можна бачити бокові межі сторінки, кінець однієї сторінки і початок іншої. Цей режим найкраще використовувати для документів на кількох сторінках.

Режим *Структура* використовується для об'ємних документів, які мають головні і другорядні розділи. При переході в цей режим зверху з'являється нова панель інструментів, за допомогою якої можна підвищувати або знижувати рівні заголовків, міняти порядок розташування розділів.

В режимі *На весь экран* у вікні Word залишається лише текст і кнопка *Вернуть обычный режим*. Режим використовується, коли потрібно якнайбільше місця для тексту, що вводиться.

Перед набором тексту потрібно встановити параметри сторінки. Для цього використовують діалогове вікно *Параметры страницы*, яке буде відкрите після команди *Файл → Параметры страницы....* В цьому вікні можна задати праве, ліве, верхнє і нижнє поля сторінки, встановити розміри аркуша та його орієнтацію тощо.

Створення текстового документа проводиться у такі етапи.

- **Набір, зберігання та редагування текстового документа.** На цьому етапі здійснюють введення тексту з клавіатури, зберігання його у вигляді файлу на магнітному диску, виправлення допущених помилок та внесення змін у текст.
- **Форматування документа** полягає у наданні документу потрібного вигляду (встановлення параметрів сторінок, шрифтів, оформлення абзаців тощо).
- **Макетування** полягає у підготовці текстового документа до друку (оформлення заголовків, розбивка на сторінки, їх нумерація, внесення у текст формул, написів та малюнків, збирання змісту, оформлення титульних сторінок і службової інформації).
- **Друк** полягає у виведенні на папір всіх або вибраних сторінок (фрагментів) створеного документа.

Набір, зберігання та редагування текстового документа

При введенні тексту слід пам'ятати та дотримуватись таких зауважень і правил.

- Символи, що вводяться з клавіатури, з'являються у місці розташування курсору клавіатури.
- Символи можуть вводитися в режимі заміни або вставки. Переключення між режимами здійснюється клавішею *Insert*.
- При наборі тексту між словами не ставити більше одного пропуску.
- Розділовий знак не відривати від слова, за яким він стоїть. Після кожного розділового знаку ставити пропуск.
- Після відкриваючих дужок або лапок зразу писати текст, закриваючі дужки або лапки писати зразу за текстом.
- До і після дефісу не ставити пропуск.
- До і після тире ставити пропуск.
- Для видалення символу зліва від курсору потрібно натиснути клавішу ← (*BackSpace*).
- Для видалення символу справа від курсору потрібно натиснути клавішу *Delete*.
- Для відміни останньої виконаної дії можна використати пункт меню *Правка → Отменить*. Якщо відміна була виконана помилково, потрібно використати пункт меню *Правка → Повторить*.
- Користувач не повинен стежити за розташуванням слів у межах абзацу. Редактор автоматично переносить текст на наступний рядок, коли поточний досяг правого кінця. Клавіша *Enter* натискається лише в кінці абзацу.
- У деяких випадках необхідно, щоб розділені пропуском слова були розташовані в одному рядку. Наприклад, ініціали та прізвище повинні завжди знаходитися в одному рядку. Щоб домогтися цього, між словами ставиться *нерозривний пропуск* (одночасно натиснути клавіші *Ctrl + Shift + Пропуск*).

Для збереження створеного документа необхідно виконати таку послідовність дій у меню:

- *Файл → Сохранить*, після чого з'явиться вікно *Сохранение документа*;
- у полі *Папка*: потрібно натиснути лівою кнопкою миші по значку ▼ і обрати ту папку, в якій буде зберігатися файл;
- у полі *Имя файла*: потрібно вказати ім'я файлу, що зберігається. За замовчуванням файлу надається ім'я

Document1.doc (або *Doc1.doc*).

- Натиснути кнопку вікна *Сохранить*.

При повторному збереженні документу (після редагування) виконання команди *Файл → Сохранить* призведе до збереження у файл з тим самим іменем у ту саму папку⁵. Команда *Файл → Сохранить как...* використовується у випадку, коли потрібно змінити ім'я документа або місце його розташування.

Для відкриття раніше створеного документа потрібно виконати таку послідовність команд у меню: *Файл → Открыть....* У вікні *Открытие документа*, що є подібним до вікна *Сохранение документа*, зазначаємо диск і папку, у якій знаходиться шуканий документ. У списку поданих файлів зазначеної папки можна відмітити потрібний і натиснути кнопку вікна *Открыть*.

До засобів редагування текстових документів належать такі:

- **Перевірка орфографії.** В процесі введення тексту виконується пошук кожного введенного слова у внутрішньому словнику. Якщо такого слова (з урахуванням дозволених змін) у словнику нема, то воно на екрані підкреслюється червоною хвилястою лінією.


- **Перевірка граматики.** При введенні тексту автоматично виконується перевірка граматики з підкресленням зеленою хвилястою лінією тексту, який не відповідає граматичним і стилістичним правилам.

- **Знайти і замінити.** Завдяки цим засобам можна знайти у багато сторінковому документі потрібний фрагмент (слово, кілька слів, речення), видалити або замінити на інший по всьому документу.

- **Російсько-український переклад окремих слів, речень і документів.**

Перевірка орфографії

Щоб перевіряти, наприклад, український текст, необхідно виконати команду: *Сервис → Язык → Выбрать язык* У вікні *Язык*, що відкриється, вибрати *Украинский* і натиснути кнопку *ОК*.

⁵ Для швидкого збереження документу можна також натиснути лівою кнопкою миші по значку  на панелі *Стандартная*

При наборі слова після натискання на пропуск або розділовий знак автоматично виконується процедура його порівняння із внутрішнім словником. Якщо слово не знайдено у словнику, воно підкреслюється червоною хвилястою лінією.

Користувач може залишити слово таким, як воно є, або виправити помилку, щоб зникла червона лінія, або звернутися за правильним словом до внутрішнього словника і замінити ним помилкове.

В останньому випадку треба клацнути правою кнопкою на підкресленому слові, з'явиться контекстне меню з одним або кількома словами, подібними до підкресленого.

Для перевірки орфографії всього документа треба виконати команду *Сервис* → *Правописание* або натиснути кнопку *Правописание* на панелі інструментів.

Почне працювати програма перевірки, яка буде послідовно читати весь текст і виводити на екран неправильні слова. У діалоговому вікні виводиться фрагмент документа з виділеною у ньому помилкою і варіанти її виправлення. Якщо документ включає фрагменти тексту, написані іншими мовами, то вони будуть показані як неправильні. Наприклад, при перевірці українського тексту слова російською і англійською мовами будуть підкреслені як помилкові.

Користувач вибирає варіант слова для виправлення або приймає інше рішення і натискає відповідну кнопку.

Перевірка граматики

Перевірка граматики — засіб, який автоматично діє при наборі текстового документа. Якщо набране речення своєю будовою порушує встановлені в Word правила, то воно буде підкреслене зеленою лінією. Щоб виправити або пропустити помилку, треба клацнути правою кнопкою миші на підкресленому зеленим тексті. З'явиться контекстне меню, у якому зверху міститься один або кілька варіантів структури речення з повідомленням про характер помилки. При виборі команди *Пропустить предложение* зелена лінія зникне і речення залишиться без змін. При виборі команди *Грамматика* відкриється діалогове вікно, зверху якого показано речення з помилкою, знизу показані його варіанти без помилки. За допомогою відповідних кнопок можна виправити речення або пропустити помилку.

Тезаурус (синоніми)

Для знаходження тезауруса (синоніма), треба виконати такі дії.

- Виділити слово, для якого шукаються синоніми.
- Виконати команду *Сервис* → *Язык* → *Тезаурус*.
- У вікні, що відкриється, із списку синонімів виберіть слово, яке найбільш відповідає змісту.
- Клацніть на кнопці *Заменить*, вікно закриється, виділене слово заміниться на вибраний синонім.

Крім режимів автоматизованої перевірки орфографії і граматики, Word пропонує ще такі засоби автоматизації:

Автозаміна — автоматична заміна із підготовленого списку слів з типовими помилками, скороченнями на правильні або повні.

Щоб ввести нові пари слів в *Автозаміну*, треба виконати такі дії.

- Виконати команди *Сервис* → *Автозамена*.
- У вікні *Автозамена*, що відкриється, у полі *Заменить* написати неправильне або скорочене слово, у полі *на:* записати його правильне або повне написання.
- Натиснути кнопку *Добавить*.

Автотекст — автоматична вставка у документ наперед підготовлених речень, абзаців і навіть кількох сторінок тексту.

Щоб ввести нові вирази в *Автотекст*, треба виконати такі дії:

- Виконати команди *Вставка* → *Автотекст* → *Автотекст...*, далі у вікні *Автозамена* відкрити вставку *Автотекст*.
- У полі *Имя элемента* внести потрібне речення.
- Натиснути кнопки *Добавить* і *ОК*.

Знайти і замінити

Щоб знайти потрібний фрагмент, необхідно:

- Подати команду *Правка* → *Найти ...*, з'явиться діалогове вікно *Найти и заменить*.
- У полі *Найти:* ввести текст-взірець.
- Для виконання пошуку натиснути кнопку *Найти далее*.

Пошук проводиться по всьому документу, починаючи від місця знаходження курсору і при бажанні з початку документа. Перший збіг фрагменту документа і взірця виділяється іншим кольором, курсор зупиниться на початку фрагменту. Якщо треба продовжити пошук, знову натиснути кнопку *Найти далее* або клавішу F4.

Для заміни одного фрагменту (символу, слова, речення, абзацу) на інший необхідно вибрати команду *Правка* → *Заменить ..* або у попередньому вікні клацнути на вкладці *Заменить*. Далі у полі *Найти:* ввести текст-взірець, клавішею Tab перейти у поле *Заменить на:* і ввести новий текст. Для пошуку тексту-взірця у документі натиснути кнопку *Найти далее*. Заміну можна проводити як із запитом на кожну дію *Заменить* або *Найти далее*, так у режимі дозволеної заміни по всьому документу (*Заменить все*).

Українсько-російський переклад

Існують різні програми для виконання перекладу. Найчастіше є потреба у українсько-російському та російсько-українському перекладі. Серед найбільш популярних таких засобів – програма Плай.

Для перекладу слова на іншу мову, треба виконати такі дії:

- Виділити слово, яке треба перекласти.
- Відкрити меню *Плай* і виконати команду *Перевод слова*.
- У вікні *Плай: Перевод слова и синонимы* у полі *Заменить на:* буде переклад слова, а також варіанти перекладу і синоніми.
- Вибрати потрібне слово або синонім.
- Натиснути кнопку *Заменить* або *Заменить все*.

- Якщо у словнику немає відповідного слова, з'являється повідомлення *Перевод слова не найден*.

Крім перекладу окремих слів. Плай здійснює переклад виділеного фрагменту або всього документа за допомогою команди *Перевод*, для чого треба виконати такі дії:

- Виділити фрагмент, який треба перекласти (для перекладу всього документа його виділяти немає потреби).
- Відкрити меню *Плай* і виконати команду *Перевод*.
- У діалоговому вікні *Плай: Опції переклада*, що з'явиться, відкрити вкладку *Перевод*.
- При необхідності встановити необхідний напрям перекладу та інші параметри.
- Натиснути кнопку *Перевести*.

В результаті цих дій відкриється новий документ з перекладом фрагменту або всього тексту, у якому окремі слова можуть бути виділені:

- *червоним кольором* — слово відсутнє у словнику, клацання правою кнопкою додає його у словник користувача.
- *зеленим кольором* — слово має варіанти перекладу, клацання правою кнопкою відкриває список варіантів для вибору.
- *синім кольором* — слово може належати до кількох частин мови, клацання правою кнопкою відкриває список варіантів для вибору.
- *світло-синім кольором* — слово має іншомовні літери.
- *бузковим кольором* — зарезервоване слово.

Форматування текстового документу

Після того, як текстовий документ набрано, йому можна надати потрібного вигляду за допомогою команд форматування.

Форматування полягає у наданні кожному абзацу текстового документа наперед заданого вигляду: певних параметрів шрифту, вирівнювання правого і лівого краю тексту, встановлення відступів і інтервалів тощо.

Word дає можливість стирати, копіювати або переносити на інше місце вибрану частину тексту — *фрагмент тексту*. Це може бути символ, слово, речення, абзац, кілька абзаців, сторінка, кілька сторінок. Для обробки фрагменту його виділяють.

Для *виділення* слова, кількох слів або речення *протягуванням*, необхідно розмістити курсор миші на першому символі, натиснувши і не відпускаючи ліву кнопку миші, протягнути курсор до останнього символу і відпустити кнопку.

Щоб *виділити кілька рядків* тексту, необхідно розмістити курсор миші на лівому полі напроти першого рядка. Коли з'явиться стрілка, натиснувши і не відпускаючи ліву кнопку, протягнути курсор миші вниз до останнього рядка і відпустити кнопку.

Щоб *виділити весь текст* документа, досить натиснути клавіші *Ctrl + 5* (на додатковій клавіатурі)

Виділити слово, рядок, абзац чи весь текст можна також *клацанням*.

Щоб *виділити слово*, необхідно розмістити курсор миші на слові і двічі клацнути. Якщо клацнути тричі, то виділиться весь абзац.

Щоб *виділити рядок тексту*, необхідно розмістити курсор миші на лівому полі напроти нього і, коли з'явиться стрілка, клацнути.

Щоб *виділити абзац*, необхідно розмістити курсор миші на лівому полі напроти одного із його рядків і, коли з'явиться стрілка, двічі клацнути.

Щоб *виділити весь документ*, необхідно розмістити курсор миші на лівому полі напроти одного із його рядків і, коли з'явиться стрілка, тричі клацнути.

Щоб зняти виділення тексту, досить клацнути на будь-якому чистому місці екрану.

Над виділеним фрагментом тексту можна виконувати такі дії.

Вырезать — фрагмент тексту розміщується в буфері обміну⁶ і видаляється зі старого місця.

Копировать — фрагмент тексту розміщується в буфері і залишається на старому місці.

Вставить — фрагмент тексту із буферу вставляється у вказане місце. Цю операцію можна виконувати скільки завгодно разів у різних місцях документа.

Удалить — фрагмент тексту видаляється.

Щоб виконати потрібну дію над виділеним фрагментом тексту, треба відкрити меню *Правка* і клацнути на відповідній команді. Також можна використовувати *контекстне меню*, що з'являється, якщо на виділеному фрагменті тексту клацнути *правою* кнопкою миші. Слід пам'ятати, що вигляд контекстного меню залежить від того місця, де воно викликається.

Шрифти

Шрифт має такі параметри.

- Вид шрифту (гарнітура).
- Написання шрифту.
- Розмір (кегель) шрифту.
- Колір шрифту.

Параметри шрифту можна встановити для одного або декількох символів, слів, речень, рядків, для всього тексту.

Щоб встановити новий шрифт для виділеною фрагменту, необхідно виконати команду *Формат → Шрифт...* або у контекстному меню викликати команду *Шрифт....* З'явиться діалогове вікно *Шрифт*, де задаються потрібні параметри.

⁶ Буфер обміну — це спеціальна тимчасова область пам'яті, у яку можна розмістити помічений фрагмент тексту для подальших операцій з ним.

У вікні *Шрифт* вибирають вид шрифту.

Це приклад шрифту Arial розміру 10 пт.

Це приклад шрифту Courier New розміру 10 пт.

Це приклад шрифту Times New Roman розміру 10 пт.

У вікні *Начертание* встановлюють написання шрифту: **жирний**, *курсив*, підкреслений. Можливі комбінації: **жирний курсив**, *курсив підкреслений*, **жирний підкреслений** і нарешті **жирний курсив підкреслений**.

Кнопки для вибору написання шрифту, як правило, є на панелі інструментів (Ж — жирний, К — курсив, Ч — підкреслений).

У вікні *Размер* можна вибрати розмір шрифту із списку від 8 до 72 пунктів. Наприклад, розмір 14 — це розмір шрифту, як у друкарської машинки. Розмір 10 — типовий книжковий розмір.

Можна встановити свій розмір від 1 до 1638 з точністю до 0,5.

Кнопки для вибору розміру шрифту, зазвичай, є на панелі інструментів. Щоб розгорнути вікно з розмірами шрифтів, досить клацнути на кнопці ▼.

Word дозволяє спеціальні ефекти з виділеним текстом: ~~Закреслений~~, ^{надрядковий}, _{підрядковий}, а також ~~подвійне закреслення~~, з тінню, контур, піднятий, утоплений тощо.

Для тексту можна також задавати колір.

Word дає можливість збільшити або зменшити інтервал між символами. Робиться це так:

- виконати команду *Шрифт*, відкрити вкладку *Інтервал*;
- щоб збільшити інтервал між символами, треба збільшити значення лічильника *На:*. У полі *Інтервал*: встановиться режим *Разреженный*;
- щоб зменшити інтервал між символами, треба зменшити значення лічильника *На:*. У полі *Інтервал*: встановиться режим *Уплотненный*;
- натиснути кнопку ОК.

Крім шрифтів з літерами, є шрифти, які мають графічні зображення. Символи такого шрифту можна вставляти в текст тільки послідовно по одному. Щоб вставити символ в текст, треба виконати такі дії:

- виконати команду *Вставка* → *Символ...* — з'явиться діалогове вікно *Символ*;
- у вікні *Символ* у списку *Шрифт* вибрати потрібний шрифт;
- знайти потрібний символ, клацнути мишею на ньому;
- натиснути кнопку *Вставити* — символ з'явиться в тексті;
- натиснути кнопку *Закрити* — зникне діалогове вікно;
- якщо потрібен ще один символ, повторити попередні дії.

Форматування абзаців

Абзац — це текст, який починається з відступу 1-го рядка від лівого краю, продовжується рядками без відступу (або з іншим відступом) і закінчується натисканням клавіші *Enter*.

Кожний абзац можна формувати окремо. Щоб вибрати абзац, досить на ньому у будь-якому місці розмістити курсор клавіатури.

Можна формувати кілька абзаців, для цього досить їх виділити

Абзац можна вирівнювати по лівому краю, по центру, по правому краю, по ширині. Щоб вирівняти текст у вибраному абзаці або групі абзаців, використовують відповідні кнопки на панелі інструментів, “швидкі” клавіші або меню *Формат* → *Абзац*.

Відступ зліва — встановлення величини відступу абзацу від лівої границі максимально можливого розміщення тексту.

Відступ справа — встановлення величини відступу абзацу від правої границі максимально можливого розміщення тексту.

Відступ 1-го рядка — встановлення зсуву вправо 1-го рядка абзацу відносно лівої границі можливого розміщення тексту.

Виступ 1-го рядка — встановлення зсуву вліво 1-го рядка абзацу відносно лівої границі можливого розміщення тексту.

Щоб встановити відступи, можна скористатися бігунками на горизонтальній лінійці і виконати такі дії:

- встановити курсор клавіатури на абзац, для якого будуть встановлюватися нові відступи;
- курсор миші встановити на потрібний бігунок;
- натиснути ліву клавішу миші, не відпускаючи її, пересунути бігунок на потрібне місце, відпустити клавішу.

Встановлювати вирівнювання, відступи зліва, справа і 1-го рядка можна також за допомогою діалогового вікна *Абзац*. Робиться це так:

- виконати команду *Формат* → *Абзац* або викликати контекстне меню, у якому виконати команду *Абзац*;
- у діалоговому вікні *Абзац*, що з'явиться, вибрати вкладку *Отступы и интервалы*;
- встановити потрібне вирівнювання у списку *Выравнивание*;;
- встановити потрібні інтервали на лічильниках *Отступ слева*:, *Отступ справа* і *Первая строка*;;
- натиснути кнопку ОК.

Щоб зробити *міжабзацні інтервали*, треба виконати такі дії:

- у діалоговому вікні *Абзац* вибрати вкладку *Отступы и интервалы*;
- встановити потрібні інтервали на лічильниках *Интервал перед*: або *Интервал после*;;
- натиснути кнопку ОК .
- Щоб встановити *міжрядкові інтервали*, треба виконати такі дії:

- у діалоговому вікні *Абзац* вибрати вкладку *Отступы и интервалы*;
- відкрити список *Междустрочный*;
- для інтервалу *Точно* встановити на лічильнику *Значение* інтервал менше 12 (стиснутий) або більше 12 (розтягнутий);
- для інтервалу *Множитель* встановити на лічильнику *Значение* коефіцієнт інтервалу між рядками менше 1 (стиснутий) або більше 1 (розтягнутий);
- натиснути кнопку *ОК*.

Щоб встановити *положення абзацу* на сторінці, треба виконати команду *Формат → Абзац* або клацнути на команді *Абзац* із контекстного меню. В діалоговому вікні *Абзац*, що з'явиться, відкрити вкладку *Положение на странице*.

На цій вкладці можна встановити такі прапорці:

Запрет висячих строк — абзац не може розриватися так, щоб на наступну сторінку переносився останній неповний рядок. Переноситься не менше двох рядків.

Не разрывать абзац — абзац завжди знаходиться на одній сторінці, не розривається на дві частини.

Не отрывать от следующего — абзац не відривається від наступного абзацу, і якщо той попадає на наступну сторінку, то за ним переноситься і перший абзац.

С новой страницы — такий абзац починається з нової сторінки незалежно від заповнення попередньої.

2. Приклад практичної роботи⁷

Набір, зберігання, редагування та форматування текстового документа

1. Завантажити програму Microsoft Word

Пуск→Программы→Microsoft Word

2. В робочій області програми набрати наступний текст (виділений рамкою):

1	Ректору ЖДУ ім. І.Франка
2	проф. Сауху П.Ю.
3	<i>Прізвище Ім'я По-батькові</i>
4	<i>Адреса проживання</i>
5	ЗАЯВА
6	Прошу зарахувати мене на курси “Оператор ПК” при
7	Житомирському державному університеті.
8	До заяви додаю:
9	копію паспорта;
10	договір;
11	копію чека.
12	“___” _____ 200__ р. <i>Прізвище Ініціали</i>

Між словами робити один пробіл.

3. Відформатувати набраний текст так, щоб він мав наступний вигляд:

Ректору ЖДУ ім. І.Франка
проф. .Сауху П.Ю.
Петренка Вадима Петровича,
що проживає в м. Житомирі по вул. Франка, 4 кім. 327

ЗАЯВА

Прошу зарахувати мене на курси “Оператор ПК” при Житомирському державному університеті.

До заяви додаю:

копію паспорта;

договір;

копію чека.

“___” _____ 200__ р.

Прізвище Ініціали

Щоб відформатувати текст необхідно:

1. Виділити абзац, який будемо формувати. (для цього необхідно поставити вертикальний курсор на абзац, який будемо формувати)
2. Вибрати в меню Word *Формат→Абзац*.
3. В діалоговому вікні *Абзац* зробити необхідні зміни і натиснути кнопку *Ок*.
Параметри для форматування абзацу наведені в наступній таблиці:

⁷ Інструкція до роботи скорочена за рахунок наочних малюнків вікон MS WORD

Номер абзацу	Вирівнювання	Відступ		Відступ першого рядка	Інтервал		Інтервал між рядками
		зліва	справа		перед	після	
1	по лівому краю	8	0	0	0	0	одинарний
2							
3							
4							
5	по центру	0	0	0	36	18	полупоторний
6	по ширині	0	0	1	0	0	множник 1,2
7							
8	по лівому краю	2	0	0	0	0	множник 1,2
9							
10							
11							
12	по лівому краю	1	0	0	24	0	одинарний

Для абзаців 1, 2, 3 та 6, 7, 8 можна виконати форматування одночасно. Для цього необхідно виділити всі три абзаци (підвести курсор мишки до лівого верхнього краю абзаців, натиснути і зафіксувати ліву кнопку миші, протягнути курсор до правого нижнього краю абзаців і відпустити ліву кнопку миші) і виконати форматування (Формат→Абзац і т.д.).

4. Змінити шрифти, розміри шрифтів та стиль шрифтів, щоб текст мав наступний вигляд:

Для всього тексту вибрати шрифт –Times New Roman, розмір –12.

Для зміни параметрів шрифту необхідно:

1. Виділити фрагмент тексту, для якого необхідно провести зміни. (для цього необхідно підвести курсор миші до лівої верхньої межі фрагменту тексту, натиснути і зафіксувати ліву кнопку миші, протягнути курсор до правої нижньої межі і відпустити ліву кнопку миші)
2. Вибрати в меню Word *Формат→Шрифт*.
3. В діалоговому вікні Шрифт зробити необхідні зміни і натиснути кнопку Ок.

Параметри для форматування тексту наведені в наступній таблиці:

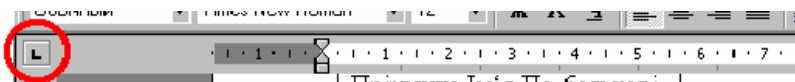
Фрагмент тексту	Стиль шрифту	Розмір шрифту
Прізвище Ім'я По-батькові	Курсив	12
Адреса проживання	Підкреслений курсив	12
ЗАЯВА	Напівжирний	18
Оператор ПК	Підкреслений	12
Прізвище Ініціали	Курсив	12

Форматування останнього фрагменту тексту, а саме:

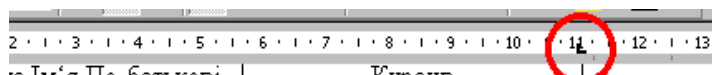
“___” _____ 200__ р. *Прізвище Ініціали*

виконують при допомозі табуляції, яку можна встановити при допомозі лінійки. Для цього необхідно виконати наступне:

- 1) Поставити курсор на абзаци (в будь-якому місці);
- 2) Зліва зверху робочої області (перетин лінійок) вибрати табуляцію по лівому краю (натисканням лівої кнопки миші на перетині лінійок можна змінити вирівнювання табуляції);



- 3) Поставити вибрану табуляцію на 11 см (натиснути на вертикальній лінійці (зверху) на нижній її частині на позначці 11); після чого на лінійці з'явиться знак табуляції;



- 4) Поставити курсор між фрагментом “___” _____ 200__ р. і фрагментом *Прізвище Ініціали* та натиснути клавішу Tab.

5. Зберегти документ в персональну папку з ім'ям text.doc (розширення файлу не вказується).

Файл→Сохранить

Далі виконуються дії, аналогічні при збереженні документів в таких редакторах, як Блокнот, WordPad, Paint.

6. Закрийте Word. Перейдіть в персональну папку та шляхом подвійного натискання на створеному файлі – text, відкрийте цей файл для редагування.

7. Доповніть документ таким текстом та відформатуйте його як наведено нижче:

Додаток 1

**Перелік програм та операційних систем,
які входять до програми вивчення на курсах:**

Операційна система Windows	8 год.
Текстовий редактор Word	18 год.
Табличний процесор Excel	16 год.
Вивчення інших прикладних та сервісних програм	6 год.
Всього	48 год.

8. Відформатуйте набраний текст по наступним правилам:

* абзац «Додаток 1» має розмір шрифту 18, напівжирний, відступи після – 24, повинен бути на новій сторінці:

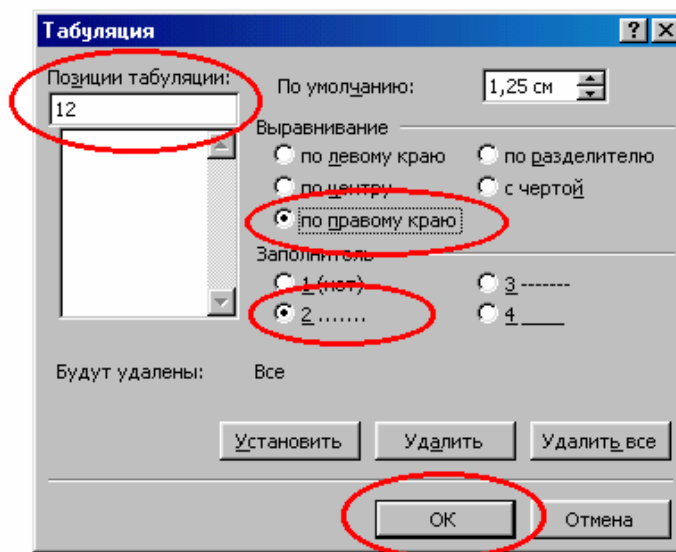
Формат→Абзац→Положення на странице→с новой страницы

* абзац «Перелік програм ...» має розмір шрифту 14, напівжирний підкреслений, відступи перед – 3, після – 24. Для того, щоб цей заголовок був в двох рядках і розривався саме після слова “систем,” потрібно після цього слова поставити м'який перенос при допомозі комбінації клавіш Shift+Enter.

* текст всіх інших абзаців – 12 розміру з відступом зліва – 2 см та відступом справа – 6 см.

* вирівнювання табуляції – справа, заповнювач табуляції, а саме: , позиції табуляції – 12 см., вказується при допомозі вікна Табуляція:

Формат→Табуляція



Доповніть документ таким текстом та відформатуйте його як наведено нижче:

Додаток 2

Програмне забезпечення курсів

Заняття проводяться три рази на тиждень о 18⁰⁰ в аудиторіях 315_а та 315_б, які оснащені наступним програмним та апаратним забезпеченням.

OC Windows 2000	OC Windows XP
MS Office 2000	MS Office XP
- MS Word 2000	- MS Word XP
- MS Excel 2000	- MS Excel XP
- MS Access 2000	- MS Access XP

9. Відформатувати текст по вимогам, що наведені нижче:

* “Додаток 2” та назва додатку мають ті ж параметри, що й ці ж заголовки в додатку 1

* 18⁰⁰, 315_а, 315_б – досягається шляхом вибору

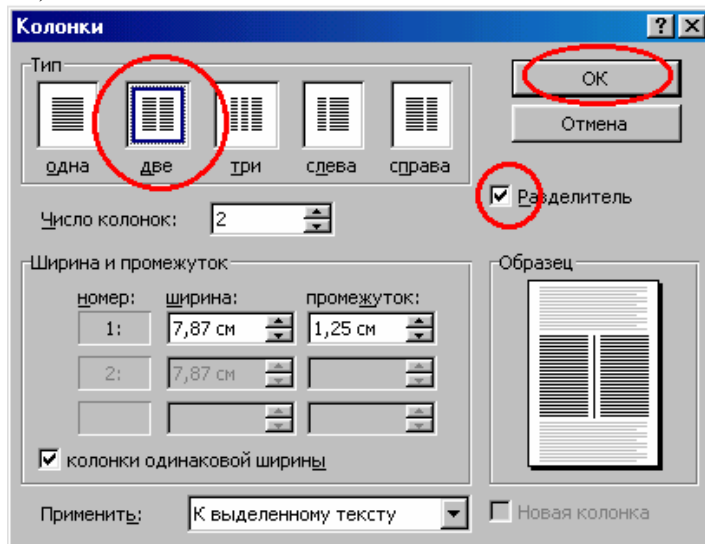
Формат→Шрифт→Верхний индекс→Ок

та

Формат→Шрифт→Нижний индекс→Ок

Текст, що відображений у вигляді двох колонок (назви апаратного та програмного забезпечення) потрібно спочатку набрати, потім виділити та виконати наступну послідовність:

Формат→Колонки→Тип - две, встановити Разделитель→Ок



* Перелік програмних продуктів, а саме:

- MS Word XP
- MS Excel XP
- MS Access XP

виконується у вигляді маркованого списку:

Формат→Список→Маркированный→Ок

10. Зберегти всі зміни в документі під тим самим ім'ям (text).

11. Закрити всі програми і завершити роботу на комп'ютері.

Розділ 2

Робота зі списками і таблицями

1. Теоретичні відомості

Використання списків

Перелік предметів, об'єктів, пункти простого чи складного плану або будь-яку іншу впорядковану інформацію зручно подавати у вигляді списків. У редакторі Word можна створити два види списків: *нумеровані* (пункти послідовно нумеруються) та *маркерні* (кожен з пунктів позначається однаковим маркером. Наприклад: —, *, • тощо).

Якщо ввести на початку абзацу цифру, а за нею крапку або круглу дужку і пропуск, то поточний і всі наступні абзаци будуть створюватися як елементи *нумерованого* списку. Замість цифр можна використовувати великі та малі літери. Наприклад: а), І), ІІ), А.

Створення списку закінчується подвійним натисканням клавіші <Enter> в кінці абзацу.

Для створення *маркерного* списку потрібно ввести на початку абзацу символ зірочки (*) або символ дефісу (–). У першому випадку маркер списку буде круглим, у другому – звичайний дефіс. Після маркера потрібно ввести пропуск, набрати текст і натиснути клавішу <Enter> – текст абзацу буде перетворено в елемент маркерного списку, а наступний абзац почнеться з маркера списку.

Щоб змінити або визначити формат списку, потрібно виконати команду у меню **Формат/Список...** У діалоговому вікні *Список* можна обрати потрібну вкладку *Нумерованный* (та визначити спосіб нумерації списку) або *Маркированный* (та обрати вид маркера).

Вкладка *Многоуровневый* дозволяє створити список, який містить декілька вкладених пунктів, що нумеруються окремо.

Для списків як і для звичайних абзацив можна встановлювати певні параметри абзацив та шрифту, скориставшись кнопкою вікна *Изменить*.

Використання таблиць

Часто текстові дані можуть подаватися у вигляді таблиць. Таблиця складається із *стовпчиків* та *рядків*. Перетин рядка та стовпчика називається *комірком* таблиці.

Word дає дві можливості для створення таблиці:

- намалювати таблицю;
- додати (вставити) таблицю.

Обидва засоби мають рівні можливості, користувач вибирає один із методів на свій смак і користується ним.

Перед створенням таблиці необхідно визначити, скільки стовпчиків і рядків вона повинна мати. Якщо у підрахунках допущена помилка, то можна на будь-якому етапі створення і заповнення таблиці додати потрібну кількість стовпчиків і (або) рядків. Якщо потрібна таблиця стандартного вигляду, краще вставити таблицю. Саме цей спосіб розглядатимемо у подальшому.

Щоб вставити таблицю, треба виконати такі дії:

- Встановити курсор клавіатури на те місце, де повинна бути таблиця.
- Виконати команду в меню *Таблиця/Добавить таблицу*.
- У діалоговому вікні *Вставка таблицы*, що з'явиться, встановити потрібну кількість стовпчиків і рядків.
- Натиснути кнопку вікна *<ОК>*, після чого на екрані з'явиться таблиця із вказаною кількістю стовпчиків і рядків.

Від початку стовпчики мають однакову ширину (відповідно, рядки – висоту). Для зміни розмірів стовпчиків (рядків) потрібно виконати наступне:

Не виділяючи комірок таблиці, перевести курсор миші на внутрішню вертикальну границю стовпчика (на горизонтальну границю рядка) з'явиться двонаправлена стрілка. Натиснувши і не відпускаючи ліву кнопку миші, перетягти стрілку для збільшення або зменшення ширини стовпчика (рядка). При перетягуванні границі з'являється штрихова лінія, яка показує, де буде проходити нова границя.

Якщо перетягування границі виконувати при виділеній одній або кількох комірках, то нова границя встановиться тільки для них із зсувом інших комірок.

Заповнення таблиці

Введення тексту в таблицю здійснюють по коміркам, причому в будь-якому порядку. Комірка може містити текст, який складається з декількох абзаців.

У кожній окремій комірці таблиці можна встановити певний шрифт (розмір, нарис тощо) та відформатувати текст комірки як текст окремого абзацу. Якщо параметри для декількох комірок однакові, то можна виокремити ці комірки та встановити потрібні параметри одразу для всіх комірок. *Введений текст у таких комірках буде розташований відповідно до параметрів форматування.*

Виокремлення елементів таблиці

Виокремлення рядка таблиці. Помістити вказівку миші за ліву межу таблиці на рівні рядка (вказівка миші прийме форму нахиленої вправо стрілки – ↗) та натиснути ліву кнопку миші – рядок виокремлено. Якщо після виокремлення рядка ліву кнопку миші не відпускати, а вертикально протягнути мишею, а потім відпустити кнопку, то це приведе до виокремлення декількох рядків таблиці.

Виокремлення стовпчика таблиці. Помістити вказівку миші за верхню межу таблиці над стовпчиком (так, щоб вказівка набула форми ↘) та натиснути ліву кнопку миші – стовпчик виокремлено. Якщо після виокремлення стовпчика ліву кнопку миші не відпускати, а горизонтально протягнути, а потім відпустити кнопку, то це дасть змогу виокремити декілька стовпчиків таблиці.

Виокремлення комірки таблиці. Перемістити вказівку миші на ліву крайню область комірки (вказівка прийме форму ↗) та натиснути ліву кнопку миші. Якщо після виокремлення комірки, не відпускаючи ліву кнопку, протягнути мишею (можна одночасно по вертикалі та горизонталі), а потім відпустити кнопку, то це дасть змогу виокремити декілька комірок таблиці.

Виокремити декілька комірок можна і за допомогою клавіатури: помістити курсор у потрібну комірку; натиснути клавішу *Shift* і, утримуючи її, за допомогою клавіш управління курсором на клавіатурі (← ↑ → ↓) визначити ті комірки, які потрібно виокремити.

Виокремлення усієї таблиці. Перемістити курсор у будь-яке місце таблиці, яку потрібно виділити. Виконати команду меню *Таблиця/Выделить таблицу*.

Щоб зняти виділення, потрібно натиснути лівою кнопкою миші у порожньому місці екрану, документа або екрану.

Форматування комірок

Команди для форматування таблиці зібрані на панелі інструментів *Таблицы и границы*. Щоб викликати цю панель інструментів на екран, досить натиснути кнопку *Таблицы и границы*, з'явиться панель інструментів. Команди правої половини панелі інструментів призначені для форматування комірок, їх призначення таке:

Добавить таблицу — викликати команду *Добавить таблицу*.

Об'єднати комірки — об'єднує в одну комірку дві і більше горизонтальних і (або) вертикальних виділених комірок.

Поділити комірки — ділить на задану кількість стовпчиків виділену комірку.

Вирівняти вміст комірок — при клацанні на кнопці ▼ відкривається список варіантів вирівнювання вмісту комірок.

Вирівняти ширину рядків — виділені рядки з різною висотою вирівнюються і набувають середньоарифметичного розміру.

Вирівняти ширину стовпчиків — виділені стовпчики з різною шириною вирівнюються і набувають середньоарифметичного розміру.

Автоформат таблиці — відкривається діалогове вікно для вибору варіантів границі і заливки.

Змінити напрямок тексту — у виділеній комірці повернути текст вертикально. Це корисно при вузьких стовпчиках.

Границі і заливка комірок

Границі і заливка виділених комірок виконується за допомогою вже відомих команд, що використовувались для абзацу. Ці команди розташовані на лівій половині панелі інструментів *Таблицы и границы*.

Виправлення помилок

Щоб виправити помилку в комірці, треба пересувати курсор по її тексту стрілочними клавішами і виправляти помилки. Щоб замінити вміст комірки, треба його стерти і набрати новий. Щоб стерти вміст комірки, треба її виділити і натиснути клавішу *Delete* або викликати контекстне меню натиснути кнопку *"Вырезать"*.

Щоб копіювати (або перенести) у комірку вміст іншої комірки, треба виконати такі дії:

- Виділити комірку, із якої копіюється (переноситься) вміст.
- Викликати контекстне меню.
- Виконати *Копировать* (*Вырезать*).
- Перевести курсор на потрібну комірку і клацнути.

- Викликати контекстне меню.
- Виконати команду *Вставити*.

Вставка стовпчиків і рядків

Щоб вставити стовпчик (або рядок), треба виконати такі дії:

- Виділити відповідний стовпчик (або рядок).
- Виконати команду *Таблиця =>Добавить*, з'явиться список.
- У списку виконати потрібну команду.

Для виділеного стовпчика ліворуч (або праворуч) з'явиться новий стовпчик. Для виділеного рядка вище (або нижче) з'явиться новий рядок.

Якщо є потреба вставити кілька стовпчиків (або рядків), треба виділити стільки ж стовпчиків (або рядків) у таблиці, і виконати попередні дії.

Зміна ширини стовпчика

Ширину стовпчика, крім вже вказаного способу, можна змінити ще так:

- Перетягти відповідний маркер на горизонтальній лінійці для збільшення або зменшення ширини стовпчика. При цьому з'являється штрихова лінія, яка показує, де буде проходити нова границя.
- Виділити стовпчик, ширину якого треба змінити, і виконати команду *Таблиця => Высота и ширина ячеек*, з'явиться діалогове вікно. Натиснути кнопку *Столбец*, ввести потрібне значення ширини і натиснути кнопку *ОК*.

Зміна висоти рядка

Висоту рядка, крім вже вказаного способу, можна змінити ще так:

- Перетягти відповідний маркер на вертикальній лінійці для збільшення або зменшення висоти рядка. При цьому з'являється штрихова лінія, яка показує, де буде проходити нова границя.
- Виділити рядок, висоту якого треба змінити, і виконати команду *Таблиця => Высота и ширина ячеек*, з'явиться діалогове вікно. Натиснути кнопку *Строка*, ввести потрібне значення висоти і натиснути кнопку *ОК*.

Якщо зміну ширини стовпчика виконувати при виділеній одній або кількох комірках, то нова границя встановиться тільки для них із зсувом інших комірок.

Зміна розміру і положення таблиці

Розміри і положення таблиці можна міняти за допомогою маркерів:

- нижній правий у вигляді двонаправленої стрілки — пропорційно змінити розміри таблиці, перетягуючи маркер;
- верхній лівий у вигляді "рози вітрів" — можна, встановивши курсор миші на маркер, перетягнути таблицю в інше місце документа.

Виконання обчислень

Якщо таблиця має числові дані у комірках і над ними треба виконати певні обчислення, слід виконати такі дії:

- Виділити комірку, де повинен бути результат обчислення.
- Виконати команду *Таблиця => Формула*, відкриється діалогове вікно, в якому можна вибрати формулу.
- "За замовчуванням" пропонується формула SUM(ABOVE), за якою обчислюється сума чисел всіх комірок вище від вибраної. Щоб підрахувати суму чисел комірок зліва, вибрати формулу SUM(LEFT).

Сортування рядків

Сортування рядків полягає у перестановці рядків відповідно до даних у вибраному стовпчику. Сортування можна проводити за зростанням (за алфавітом) або за зменшенням (проти алфавіту). Щоб провести сортування, треба виконати такі дії:

- Виділити комірки ОДНОГО стовпчика, згідно даних якого буде проводитися сортування.
- Виконати команду *Таблиця => Сортировка*, відкриється діалогове вікно *Сортировка текста*.
- Вибрати *По возрастанию*, якщо сортування проводиться за зростанням (за алфавітом) або *По убыванию*, якщо сортування проводиться за зменшенням (проти алфавіту).
- Натиснути кнопку *ОК*.

2. Приклад практичної роботи **Робота зі списками і таблицями**

1. Завантажити програму Microsoft Word.
2. Відкрити файл text.doc, що знаходиться в персональній папці.

Файл→Открыть

3. Перейдіть в кінець документу та додайте наступний текст:

Додаток 3

Програма курсу

1. Будова ПК. Периферія ПК. Робота з клавіатурним тренажером
 2. Поняття ОС. Види ОС:
 - Windows XP;
 - Windows NT;
 - Workstation 4.0;
 - Server 4.0;
 - Windows 2000:
 - Professional;
 - Server;
 - Unix;
 - Linux:
 - SCO;
 - RedHat;
 - BackCat;
 - FreeBSD.
 3. Файли, папки, диски, дерево каталогів.
 4. Текстовий редактор "Блокнот". Графічний редактор Paint. Текстовий редактор WordPad.
 5. ОС Windows.
 6. Пакет Microsoft Office.
 - Текстовий редактор Microsoft Word;
 - Електронна таблиця Microsoft Excel;
 - СУБД Microsoft Access;
 7. Комп'ютерна мережа.
 8. Антивірусні програми. Архіватори.
4. Відформатувати текст наступним чином:

Абзац	Вирівнювання	Відступ		Відступ першого рядка	Інтервал		Інтервал між рядками	Стиль та розмір шрифту
		зліва	справа		перед	після		
Додаток 3	по центру	0	0	0	0	6	одинарний	Напівжирний, 16
Програма курсу	по центру	0	0	0	6	6	полуторний	Напівжирний 14
Список 1,2... Список зі ➤ Список зі —	по лівому краю	1,06 ; 2,33 ; 3,17	0	0	6	0	множ. 1,1	Звичайний, 12

Щоб відформатувати текст так як зображено, необхідно (у вигляді списку):

1. Виділити абзац, який будемо форматувати. (для цього необхідно поставити вертикальний курсор на абзац, який будемо форматувати)
2. Вибрати в меню Word *Формат*→*Список*.
3. В діалоговому вікні Список вибрати необхідний тип списку: нумерований, маркірований, багаторівневий, а також необхідний символ списку (для маркірованого) чи шрифт (для нумерованого) і натиснути кнопку *Ок*.

Розділ 3

Робота з об'єктами

1. Теоретичні відомості

У текстовому документі Word можуть бути розташовані малюнки, створені у середовищі Word, графічними редакторами Windows, написи, формули та інші об'єкти.

При роботі у середовищі Word можна не турбуватися про формат графічного файлу, необхідні перетворення форматів виконуються автоматично. Крім того, є можливість вибрати тип файлу із списку *Тип файлів*. Цей список стає доступним після виконання команд *Вставка* => *Рисунок*, далі у підменю вибрати *Із файла* і в діалоговому вікні *Тип файла* вибрати потрібний тип файлу.

Зміна розмірів малюнка

Щоб змінити розміри малюнку, треба виконати такі дії:

Активізувати малюнок, для чого клацнути на ньому, біля малюнка з'являться маркери. Коли встановити курсор миші на один з маркерів, з'являється двонаправлена стрілка.

Щоб пропорційно збільшити (або зменшити) розміри малюнку, необхідно встановити курсор миші на кутовий маркер, натиснути і, не відпускаючи ліву кнопку миші, перетягти маркер від центру (або до центру) і при досягненні потрібних розмірів відпустити.

Щоб розтягнути (або стиснути) розмір в одному напрямку, необхідно встановити курсор миші на середній маркер з потрібної сторони, натиснути і не відпускати ліву кнопку миші, перетягти маркер від центру (або до центру) і при досягненні потрібних розмірів відпустити.

Переміщення малюнка

Для переміщення малюнка в інше місце виконати такі дії:

Активізувати малюнок, для чого на ньому клацнути.

Встановити курсор миші на малюнок, натиснути і не відпускати ліву кнопку миші, перетягти малюнок на потрібне місце.

Копіювання малюнка

Для копіювання малюнка виконати такі дії:

- Активізувати малюнок, для чого на ньому клацнути.
- Клацнути правою кнопкою на малюнку, у контекстному меню, що з'явиться, виконати команду *Копіювати*.
- Перевести курсор в інше місце документа. Клацнути правою кнопкою на ньому, у контекстному меню, що з'явиться, виконати команду *Вставити*.
- Малюнок, як правило, копіюється біля попереднього. Перетягнути копію малюнка у потрібне місце.

Малюнок у тексті

Малюнок і текст мають різну природу. Для сумісного відображення малюнка в тексті потрібно виконати деякі дії.

При вставці малюнка у вибране місце текстового документа "за замовчуванням" текст розсувається і знаходиться зверху і знизу малюнка. Щоб зробити інше, виконати такі дії:

Активізувати малюнок, для чого клацнути на ньому. Клацнути правою кнопкою на ньому, у контекстному меню, що з'явиться, виконати команду *Формат рисунка* У діалоговому вікні *Формат рисунка* вибрати вкладку *Положення*, у якому встановити відповідне поле *Обтекание* для сумісного існування тексту й малюнка. Натиснути кнопку *ОК*, щоб встановити вибрані параметри.

Напис у тексті

Напис — це абзац тексту, оформлений як графічний елемент, який можна розташувати у тексті, пересувати по документу, міняти його розміри тощо. Щоб створити напис, необхідно виконати такі команди:

- Розгорнути панель інструментів *Рисование*.
- Натиснути кнопку *Надпись*, перевести курсор миші на екран.
- Встановити курсор миші, що став у формі, в один із кутів майбутнього напису, натиснути на ліву кнопку і, не відпускаючи, перетягти курсор у протилежний кут напису. Одержимо прямокутник.
- Перевести курсор усередину прямокутника, набрати текст із потрібним шрифтом і вирівнюванням.

Щоб редагувати напис, треба виконати такі дії:

- Зробити напис активним, клацнувши на ньому.
- Викликати контекстне меню, виконати в ньому команду *Формат надписи...* — з'явиться діалогове вікно *Формат надписи*.
- Відкрити вкладку *Цвета и линии*, у якій встановити колір заливки, колір, шаблон, тип і товщину лінії рамки.
- Відкрити вкладку *Положение*, у якій вибрати вид обтікання для сумісного існування тексту й напису.
- При необхідності міняти розміри й вигляд рамки напису за вищезгаданими правилами.

Розмір шрифту тексту при зміні розмірів рамки напису не змінюється. Вид, написання, розмір, колір та видозміни шрифту встановлюються окремо за відомими правилами.

Вставка формул

Word дозволяє вставляти в текстовий документ різноманітні формули, які використовуються і обробляються як графічні елементи. Щоб вставити в документ формулу, треба виконати такі дії:

- Встановити курсор клавіатури на те місце в документі, де повинна бути формула.
- Виконати команду *Вставка => Объект...*, відкриється вікно *Вставка объекта*.
- Клацнути у вікні *Вставка объекта* на вкладці *Создание*.
- У списку *Тип объекта* вибрати *Microsoft Equation 3.0* і натиснути кнопку *ОК*. З'явиться макет формули і меню для вибору типу формули і її елементів.
- Вибирати в меню елементи і вставляти їх в макет формули, вводити знаки змінних і констант до одержання потрібної формули.

При необхідності перемістити або змінити розміри одержаної формули, як графічного елемента.

Малювання

Для роботи з малюнками *Word* має панель інструментів *Рисование*, за допомогою якої можна створювати малюнки, використовуючи прямі і криві лінії, прямокутники і еліпси, замальовувати їх різними кольорами, робити написи тощо.

На панелі інструментів *Рисование* у списку *Действия* серед інших є такі команди:

Группировать — групування кількох активних графічних об'єктів в один графічний об'єкт.

Разгруппировать — розгрупування одного графічного об'єкту на окремі графічні об'єкти, із яких він був згрупований.

Порядок — встановлення порядку розташування малюнка на фоні інших малюнків (на передній план, на задній план тощо).

Сетка— прив'язка малюнків до сітки, щоб їх вирівняти.

Сдвиг — зсув малюнка вправо, вліво, вгору або вниз.

Вирівнять/распределить — вирівнювання по центру або по одному із країв горизонтально або вертикально.

Повернуть/отразить — обертання малюнка на 90° або його симетричне відображення.

Малюнки складаються з окремих графічних елементів: ліній, прямокутників, кіл, написів тощо. Кожен елемент переміщається по екрану незалежно від інших. В таких умовах малюнок, як множина простих графічних елементів, буде "розлазитись" на складові частини. Щоб створити єдиний малюнок, треба об'єднати (згрупувати) його складові частини. Вибір об'єктів для створення малюнка робиться так:

- Натиснути кнопку (*Выбор объектов*).
- Перевести курсор миші в лівий верхній кут малюнка.
- Утримуючи натиснутою ліву кнопку миші, перевести курсор миші в правий нижній кут малюнка, відпустити кнопку, всі об'єкти малюнка стануть активними.
- Відкрити меню *Действия* і виконати команду *Группировать*, об'єкти об'єднуються у малюнок, який можна обробляти як єдине ціле: копіювати, переносити, змінювати його розміри тощо.

Після групування необхідно знову натиснути на кнопку *Выбор объектов*, щоб вона погасла.

Якщо клацнути на символі ▼ праворуч від кнопки *Автофигуры*, відкриється список для вибору найбільш поширених видів геометричних фігур: ліній, основних фігур, фігурних стрілок, блок-схем, зірок і стрічок, виносков. Кожен вид має справа знак ►, тобто розгортається у більш детальне меню. Щоб вибрати потрібну фігуру, треба встановити на ній курсор миші і перетягти її в текст, на екрані з'явиться знак +. Щоб одержати фігуру потрібного розміру, треба, натиснувши і утримуючи ліву кнопку миші, розтягнути фігуру.

На правій частині панелі інструментів *Рисование* є такі кнопки для виконання дій для створення малюнка:

Колір заливки

Цей засіб має дві можливості. Якщо натиснути кнопку з відерцем, то *виділений* об'єкт зафарбується у колір, що під відерцем. Якщо ж натиснути кнопку ▼ справа від відерця, відкриється діалогове вікно з палітрою у 40 кольорів. Досить клацнути на потрібному кольорі, і виділений об'єкт зафарбується у цей колір. При потребі можна підібрати відтінок кольору, клацнувши на кнопці *Дополнительные цвета заливки ..*". Для декоративної заливки слід натиснути кнопку *Способы заливки...*, відкриється меню *Заливка*, у якому можна вибрати види заливки. Щоб зняти заливку, треба натиснути кнопку *Нет заливки*.

Колір лінії

Ця кнопка дозволяє вибрати колір для *виділеної* лінії. Цей засіб має такі ж можливості, як засіб *Колір заливки*. Щоб зробити лінію невидимою, треба натиснути кнопку *Нет линий*.

Тип лінії

Ця кнопка дозволяє для намальованої і *виділеної* лінії або стрілки вибрати її тип (від тонкої 0.25 пт до товстої 6 пт, кілька видів подвійної, потрібна) із списку, що відкривається.

Тип штриха

Ця кнопка дозволяє для намальованої і *виділеної* лінії або стрілки вибрати тип штриха із списку, що відкривається.

Вид стрілки

Ця кнопка дозволяє для *виділеної* стрілки вибрати варіант із списку, що пропонується.

2. Приклад практичної роботи

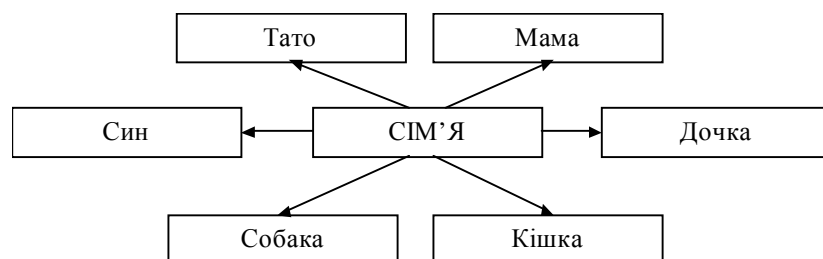
Робота з об'єктами

1. Завантажити програму Microsoft Word.
2. Відкрити файл text з вашої персональної папки.
3. До існуючого тексту додати наступний текст:

Додаток 5

Приклад екзаменаційного завдання №1

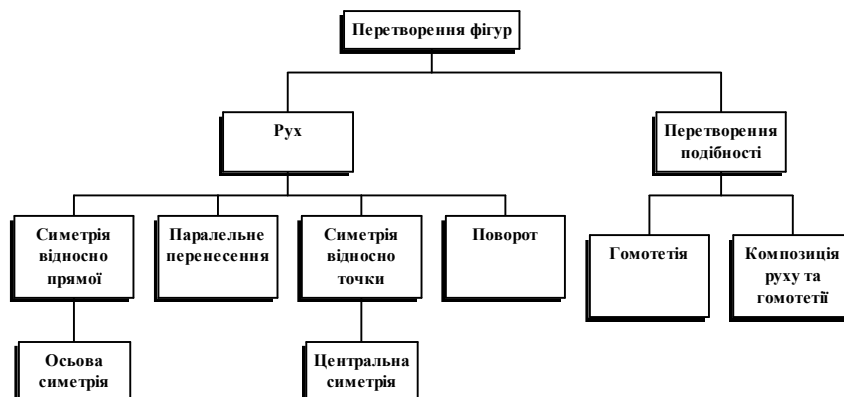
Нижче намалювати наступний рисунок:



Щоб вставити в документ рисунок, необхідно:

1. Виконати в меню Word *Вставка*→*Объект*→В діалоговому вікні "*Вставка объекта*" вибрати "*Рисунок Microsoft Word*"→*Ок*.

- У вікні редагування рисунка намалювати рисунок, використовуючи панель *Рисование*, на якій можна вибрати необхідні елементи (*Надпись, Прямоугольник, Стрелка, Линия* тощо).
- Після закінчення редагування малюнку натиснути на панелі *Изменение рисунка* кнопку *Восстановить границы рисунка* і лише потім натиснути кнопку *Заккрыть рисунок*.
- Побудувати наступну структуру даних:



Щоб вставити організаційну структуру даних, необхідно:

- Виконати в меню Word *Вставка→Объект→В* діалоговому вікні "*Вставка объекта*" вибрати "*MS Organization Chart 2.0*"→*Ок*.
- У вікні редагування організаційної діаграми, використовуючи панель *MS Organization Chart* та пункт меню "*Стиль*", створити необхідну структуру даних.
- Після закінчення редагування діаграми закрити MS Organization Chart.

- Набрати наступну формулу:

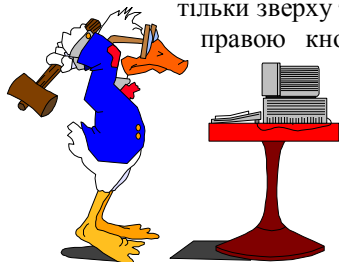
$$R = \frac{x^2(x+1)}{b} - \sin^2(x+a) + \left| \frac{3a}{\alpha + \beta} \right| \times \sqrt{\frac{x\lambda}{\mu}} + \sum_{i=1}^{(1+n)^2} F_i^{kp}$$

Щоб написати математичну формулу, необхідно:

- Виконати в меню Word *Вставка→Объект→В* діалоговому вікні "*Вставка объекта*" вибрати "*Microsoft Equation 3.0*"→*Ок*.
- У вікні редагування формули, використовуючи панель "*Формула*" набрати необхідну формулу.
- Після закінчення редагування формули потрібно вийти з редактора формул – натиснути лівою кнопкою миші на будь-якому місці документу.

- Вставити в документ малюнок з текстом:

Цей приклад показує як можна використовувати малюнок (це відноситься до будь-якого об'єкту) разом з текстом. Текст може бути наскрізний, тобто йти поверх малюнка, може обтікати малюнок (так як в нашому прикладі) з різних сторін, може бути тільки зверху та знизу малюнка та т. ін. Всі ці можливості можна переглянути та встановити при натисненні правою кнопкою на малюнку та виборі пункту *Формат рисунка* та закладки *Обтекание*, де і встановлюються необхідні параметри.



Щоб вставити в документ малюнок, необхідно:

- Виконати в меню Word *Вставка→Рисунок→Картинки→В* діалоговому вікні "*Вставка картинки*" вибрати необхідний малюнок→*Ок*.
- Щоб розмістити текст з малюнком так, як показано в завданні, необхідно натиснути правою кнопкою миші на малюнку та вибрати "*Формат рисунка*" →

закладка *Положение*→*По контуру*

Щоб вставити в документ виноску (так як на слові "малюнок"), необхідно:

- Виконати в меню Word *Вставка→Сноска→В* діалоговому вікні вибрати "*обычную*" "*автоматическую*" → *Ок*.
- Текст виноски повинен бути набраний таким же видом шрифту, як і шрифт основного тексту, а розмір повинен бути на 2 пункти менший від розміру основного тексту.

Розділ 4

Форматування документів складної структури

1. Теоретичні відомості

Стилі

Встановлення стилів

Документ повинен мати однакові стилі оформлення тексту на всіх сторінках (шрифти і вирівнювання заголовків, відступи і інтервали абзаців, маркіровані і нумеровані списки, виділення окремих фрагментів тощо). Робити це "вручну" потребує багато уваги і часу. Для виконання такої роботи краще використовувати стилі.

Стиль встановлюється для всього абзацу, для кількох виділених абзаців або для всього документа.

Для нового документа Word пропонує набір стилів, яким можна скористатися, клацнувши на значку ▼ поля *Стиль* на панелі інструментів. Відкриється список стилів: від *Заголовок 1* — тип шрифту Arial, жирний, вирівнювання по лівому краю, розмір 14 пт до *Основной шрифт абзаца* — тип шрифту Times New Roman, звичайний, вирівнювання по лівому краю, розмір 10 пт.

Щоб встановити стиль, необхідно встановити курсор клавіатури на потрібний заголовок або абзац, відкрити список і клацнути на потрібному стилі. Текст абзацу набуде вигляду згідно вибраного стилю.

Щоб встановити єдиний стиль для кількох абзаців або всього документа, треба їх попередньо виділити і виконати вищевказані дії.

Стилі заголовків

Стилі заголовків розділів, підрозділів і т.д., повинні бути оформлені однаково по всьому документу і забезпечувати створення змісту документа. Стилі заголовків можна встановлювати за допомогою поля *Стиль* на панелі інструментів, за допомогою швидких клавіш, які створює користувач, або після виконання команди *Формат => Стиль*.

Останній засіб використовують, коли треба не тільки встановити, але й змінити вибраний стиль заголовку. Встановлення для заголовків розділів, підрозділів і т.д. стилів *Заголовок 1*, *Заголовок 2*, *Заголовок 3* дозволить автоматично зібрати зміст документа.

Ознакою некваліфікованої роботи є встановлення стилів "Заголовок" для абзаців, які не повинні виноситися у зміст.

Створення "швидких клавіш"

Користування найбільш вживаними стилями можна значно спростити, створивши для кожного з них "швидку клавішу" — одночасне натискання комбінації клавіш: Ctrl + (буква або цифра) або Alt + (буква або цифра).

Щоб встановити "швидку клавішу", треба виконати такі дії:

- Встановити курсор клавіатури на абзац, для стилю якого треба створити "швидку клавішу".
- Виконати команду *Формат => Стиль => Изменить ... => Клавиша*, відкриється діалогове вікно *Настройка клавиатуры*.

- У діалоговому вікні, що відкриється, одночасно натиснути і відпустити клавіші Ctrl (Alt) + буква або цифра, яка б відображала призначення стилю (наприклад, для стилю *Заголовок 1* можна встановити "швидкі клавіші" Ctrl + Alt + 1).

- Натиснути кнопку *Назначить*, далі на кнопці *Закреть*.
- Натиснути кнопку *ОК*, щоб повернутися у вікно *Стиль*, і у ньому на кнопці *Применить*.

Всі стилі, які використовуються в документі, зберігаються у файлі цього документа.

При перенесенні файлу документа на інший комп'ютер всі його стилі і засоби роботи з ними зберігаються.

Щоб встановити потрібний стиль для вибраного абзацу, досить встановити на нього курсор і натиснути задану комбінацію клавіш.

Встановлення колонтитулів

Колонтитул — це зарезервоване вільне місце зверху і (або) знизу сторінки для зазначення номера сторінки, назви розділу, книги тощо. Для встановлення колонтитулу, потрібно виконати такі дії:

- Вибрати команду *Вид => Колонтитулы*, з'явиться панель інструментів *Колонтитулы* і документ перемикнеться у режим роботи з колонтитулами. Текст документа зробиться блідим, і з ним працювати буде неможливо.

- Область колонтитула обведена пунктирною лінією. В цій області, починаючи із знаку *і*, записується текст колонтитула.

- Після внесення текстів колонтитулів натиснути кнопку *Закреть* — і на всіх сторінках з'являться встановлені колонтитули.

Нумерація сторінок

Для нумерації сторінок документа потрібно виконати такі дії:

- Виконати команду в меню *Вставка => Номера страниц ...*, після чого відкриється діалогове вікно *Номера страниц*.

- Зазначаємо положення (у полі *Положение*) та вирівнювання (у полі *Выравнивание*) номеру сторінки на листку та вказуємо необхідність нумерації першої сторінки.

Для визначення формату номера сторінки (арабські, латинські цифри або літери; шрифт та його розмір; зазначення номеру сторінки, з якого починається нумерація) потрібно скористатися кнопкою вікна *Формат*.

Для зміни нумерації сторінок треба виконати попередні дії, а у вікні *Формат номера страницы* встановити нові значення. Така операція можлива, якщо не міняти параметри полів *Положение* і *Выравнивание* (інакше поруч з існуючою нумерацією одержимо ще одну).

Для відміни нумерації сторінок, треба виконати такі дії:

- На будь-якій сторінці двічі клацнути на номері сторінки — появиться колонтитул.
- Знову клацнути на номері сторінки — у колонтитулі навколо номера сторінки появляться маркери у вигляді чорних квадратиків.

- Натиснути клавішу *Delete*.
- Натиснути кнопку *Закри́ть* панелі *Колонтитулы*.

Виноски

Виноски можуть бути розташованими в кінці кожної сторінки або в кінці всього документа. Щоб зробити виноску, треба виконати такі дії:

- Встановити курсор клавіатури в кінці слова, до якого робимо виноску.
- Виконати команду в меню *Вставка => Сноска....*
- У діалоговому вікні "*Сноски*", що з'явиться, встановити потрібні параметри у полях *Вставити сноску* і *Нумерація*.

• Натиснути кнопку вікна *OK* — у тексті на місці курсору клавіатури з'явиться порядковий номер виноски, а сам курсор переміститься в кінець сторінки (документа) для введення тексту виноски.

- Набрати текст виноски, після чого можна продовжувати набір тексту основного документа.

Автоматичне створення змісту

Автоматичне створення змісту стає можливою за умови використання стилів "*Заголовок*" в документі.

Для створення змісту потрібно виконати такі дії:

- Встановити курсор клавіатури в кінці або на початку документа, де повинен знаходитись зміст.
- Виконати команду у меню *Вставка => Оглавление и указатели*, після чого відкриється діалогове вікно *Оглавление и указатели*.

• Обрати вкладку "*Оглавление*".

Через деякий час на місці курсору клавіатури з'явиться зміст документа, у список стилів буде додано стилі "*Оглавление 1*", "*Оглавление 2*" і т. д. для відповідних рівнів змісту.

При встановленні на зміст курсор миші набуде вигляду руки. Досить клацнути на потрібній назві розділу або підрозділу, як на екрані з'явиться його текст.

2. Приклад практичної роботи

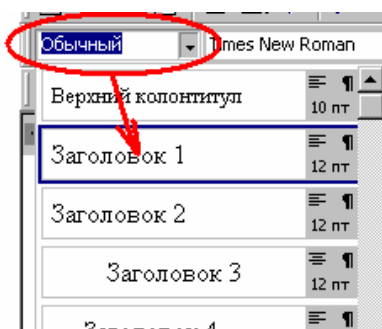
Форматування документів складної структури

Завантажити програму Microsoft Word.

1. Відкрити робочий файл text з персональної папки.
2. Застосувати до заголовків стилі. Застосування заголовків до фрагментів тексту повинно бути зроблене наступним чином:
 - заголовок 1-го рівня: ЗАЯВА;
 - заголовок 2-го рівня: Додаток 1, Додаток 2 і т.д.;
 - заголовок 3-го рівня: Назва додатку.

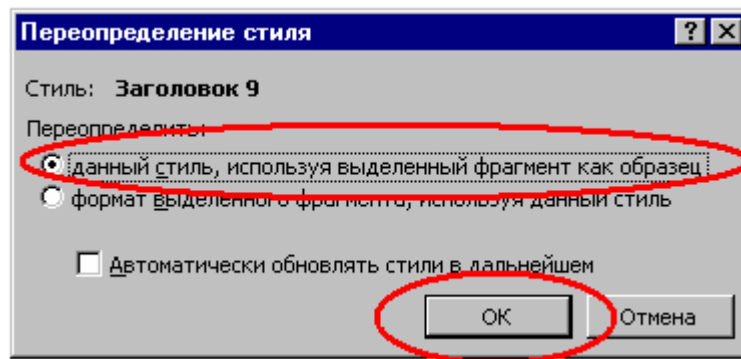
Щоб застосувати стилі до тексту, необхідно:

1. Виділити абзац, до якого потрібно застосувати стиль.
2. На панелі інструментів "*Форматирование*" вибрати з існуючого списку стилів, вибрати необхідний⁸, наприклад – Заголовок 1.



3. Виділений абзац зміниться після виконання попереднього пункту. Потім необхідно виконати зміни, необхідні для цього абзацу (вимоги – в таблиці).
4. Після закінчення редагування абзацу знову на панелі вибрати назву того ж стилю, що й в перший раз (Заголовок 1). Після чого з'явиться діалогове вікно "*Переопределение стиля*" в якому з двох можливих варіантів Переопределить: "*данный стиль, используя выделенный фрагмент как образец*"; "*формат выделенного фрагмента текста, используя данный стиль*", вибираємо перший.

⁸ Можна також створити свій власний стиль, наприклад для основного тексту і вказати самостійно його ім'я



5. В подальшому, при необхідності застосування стилю, який був раніше перевизначений, необхідно виділити необхідний абзац та вибрати цей шрифт на панелі інструментів.

№ рівня заголовку	Вимоги
1	Шрифт Times New Roman, розмір – 20, напівжирний, інтервал після – 12
2	Шрифт Arial, розмір – 18, напівжирний, інтервал після – 12
3	Шрифт Times New Roman, розмір – 16, напівжирний курсив, інтервал перед – 6, після – 12

3. Вставити нумерацію сторінок для документу (внизу сторінок по центру).

Щоб проставити нумерацію сторінок, необхідно:

1. Виконати в меню Word *Вставка*→*Номера страниц...*→В діалоговому вікні “Номера страниц” вибрати положення – внизу сторінки, вирівнювання – від центру. Самостійно проглянути можливості формату номеру сторінки.
2. Після закінчення встановлення параметрів номерів сторінок натиснути Ок.

4. Вставити наступні колонтитули для тексту: для першої сторінки – “Документ”, для всіх інших – “Додатки”.

Щоб вставити колонтитули в текст, необхідно:

1. Якщо потрібно для різних сторінок вставити різні колонтитули (так як у нашому випадку), то перед тим, як вставляти колонтитули, потрібно розбити документ на різні розділи.
2. Для вставки розриву розділу в поточному положенні, необхідно виконати в меню Word *Вставка*→*Разрыв*→В діалоговому вікні “Разрыв” вибрати новий розділ «со следующей страницы».
3. Виконати в меню Word *Вид*→*Колонтитулы* в полі редагування колонтитулу (верхнього) ввести його назву (Документ та Додаток відповідно до розділів).
4. Відформувати колонтитул – зробити шрифт курсивом з вирівнюванням справа.
5. Закрити поле редагування колонтитулу – натиснути “Закреть” на панелі “Колонтитулы”.

5. Для всього тексту документу встановити мову – Українську, та при допомозі програми Рута-Плай виправити помилки, якщо такі мають місце. Поставити автоматично перенесення слів.

Щоб перевірити документ на присутність помилок в тексті, необхідно:

1. Виділити весь текст документу.
2. Виконати в меню Word *Сервис* → *Язык* → *Выбрать язык* → В діалоговому вікні “Язык” вибрати “Украинский” → Ок. (при цьому повинно бути встановлено *Сервис* → *Параметры* → *Правописание* → *автоматически проверяют орфографию*).

Щоб слова в тексті мали автоматичний перенос, необхідно:

1. Виконати в меню Word *Сервис* → *Язык*→ *Расстановка переносов*→В діалоговому вікні “Расстановка переносов” вибрати “Автоматическая расстановка переносов”→Ок.

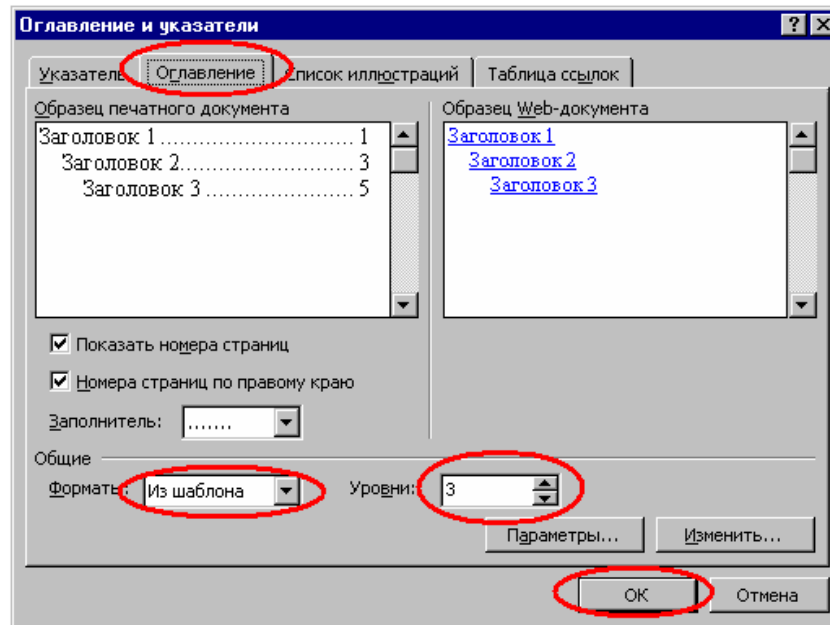
6. Вставити в кінець тексту зміст документу

Перед тим, як вставити зміст наберіть слово ЗМІСТ з нової сторінки та застосуйте до нього стиль “Заголовок 1”. Під цим заголовком поставте курсор – місце, куди буде вставлений зміст.

Щоб вставити автоматично зміст документу, необхідно:

1. По-перше, в документі повинні бути застосовані стилі!!! (див. п.3)

2. Виконати в меню Word Вставка → Оглавление и указатели → закладка Оглавление → вибрати вид – “Из шаблона”, встановити кількість рівнів = 3 → Ок



7. Зберегти всі зміни в файлі під іменем “pr_word.doc” (розширення файлу не вказується) в своїй персональній папці. Всі попередні версії цього документу знищити. Закрити всі програми і завершити роботу на комп’ютері.

Завдання до лабораторної роботи

1. Повторити теоретичний матеріал з курсу інформатики по текстовому редактору MS Word.
2. Виконати та проаналізувати практичну роботу.
3. В ході аналізу сформулювати всі знання та вміння, що формуються цією практичною роботою.
4. Розробити фрагменти конспектів уроку пояснення нового матеріалу з теми по кожному розділу, враховуючи знання і вміння, якими повинні оволодіти учні при вивченні MS Word.
5. Розробити систему контрольних питань та завдань для перевірки матеріалу теми.
6. Оформити звіт по лабораторній роботі.

ЛАБОРАТОРНА РОБОТА №11 (2 год.)

Методика вивчення графічного редактора

ОСНОВНА МЕТА	Методичні особливості вивчення графічного редактора в школі.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ ППЗ	Знати основні принципи, форми, прийоми та засоби вивчення графічного редактора. Графічний редактор MS Paint.

1. Методичні рекомендації

Подання інформації у вигляді схем, діаграм, графіків, фільмів часто є зручним для її аналізу і опрацювання. Прикладне програмне забезпечення сучасних комп'ютерів містить потужні системи опрацювання і подання графічної інформації. Вони використовуються в діловій графіці, при розв'язуванні прикладних задач, в навчаючих програмах, для створення мультиплікаційних фільмів, комп'ютерних ігор тощо. Такі програмні засоби дають змогу подавати окремі фрагменти рисунків у збільшеному вигляді, демонструвати перерізи тіл, вносити зміни до графічної інформації, зберігати її на зовнішніх носіях, виводити на друкуючі пристрої тощо.

Для створення і редагування графічних зображень на екрані дисплея існують спеціальні програми, які називають *графічними редакторами*. У середовищі Windows таким редактором є Paint. Аналогічний редактор Paint Brush використовується в операційній системі MS DOS.

Розрізняють три види комп'ютерної графіки: *растрову, векторну і фрактальну*. Вони розрізняються принципами формування зображень на екрані монітора та при роздрукуванні на папері. Основним елементом растрового зображення є точка (піксель – *Picnute Element*). Кількість точок у графічному зображенні залежить від типу монітора та роздільної здатності встановленого режиму роботи. Зображення растрової графіки або будують вручну (використовуючи інструменти та можливості сучасних редакторів), або утворюють електронні копії зображень за допомогою цифрових фото- і відеокамер. Більшість редакторів растрової графіки орієнтовані не стільки на створення, скільки на їх подальшу обробку. В Інтернеті використовуються поки тільки растрові зображення. Недоліками растрових зображень є те, що вони на дисках займають великі об'єми пам'яті і їх неможливо збільшувати для розгляду окремих деталей (в результаті збільшення точки стають крупнішими). Основним елементом зображення векторної графіки є лінія. Об'єм пам'яті, який займає лінія, не залежить від її розмірів, оскільки лінії подаються за допомогою деякої кількості параметрів, які містять її математичний опис. На основі ліній будують інші об'єкти, простіші об'єкти об'єднуються в більш складніші. Одні об'єкти, незалежно від інших, можна редагувати, переміщати, повертати, масштабувати. Через такий підхід векторну графіку називають об'єктно-орієнтовною. Векторну графіку в основному використовують для створення графічних ілюстрацій. У випадку фрактальної графіки зображення автоматично генеруються на основі математичних розрахунків. Створення фрактальної художньої композиції зводиться до її програмування. Її часто використовують у розважальних програмах.

Графічний редактор Paint входить до складу стандартних програм Windows і призначений для створення і обробки малюнків растрової графіки.

Для запуску редактора слід виконати команду *Пуск → Программи → Стандартные → Графический редактор Paint*. Після запуску на екрані з'являється вікно редактора.

Рядок заголовку містить назву програми *Paint* і ім'я графічного файлу, який редагується. Рядок меню розміщується під рядком заголовку і включає такі пункти:

Файл – робота з графічними файлами (створення, відкриття, збереження, друкування).

Правка – редагування малюнка.

Вид – управління зображенням деяких елементів вікна.

Рисунок – обробка малюнка (встановлення розмірів, поворотів).

Параметри – завантаження, зміна і збереження палітри кольорів.

? – виклик довідкової інформації.

Кожний пункт меню має підменю.

Для відкриття меню слід натиснути клавішу *Alt* або *F10*. Після цього один з пунктів меню виділиться інверсним кольором. Для виділення потрібного пункту меню слід користуватися клавішами горизонтального переміщення курсору. Відкривається виділений пункт меню при натискуванні клавіші *Enter*. Відкрити меню і підменю зручніше за допомогою миші, встановивши курсор на потрібному пункті меню і натиснувши ліву кнопку.

Панель інструментів для малювання розміщена ліворуч і включає ряд кнопок, за допомогою яких можна вибрати необхідний інструмент, побудувати типові елементи малюнка і т. д. При фіксації курсору миші на кнопці під нею з'являється назва, а в рядку стану – коротка довідка про призначення кнопки.

Палітра кольорів розміщена внизу і включає 28 зафарбованих різними кольорами квадратів. Із цих кольорів користувач може вибрати основний колір, яким виконуються побудови, і колір фону.

Область малювання призначена для побудови малюнка. Вона обмежена головним меню, панеллю інструментів, палітрою. Розміри області малювання встановлюються командою *Рисунок → Атрибуты*.

Виведенням на екран деяких елементів вікна Paint можна керувати за допомогою команд-перемикачів меню *Вид*.

Почати вивчення матеріалу слід з пояснення принципів відображення графічної інформації на екрані дисплею. Головне – дати учням зрозуміти, що відображується на екрані картинка зберігається в ОЗП в такому ж вигляді, як і будь-яка

інша інформація (числова або літерна), але якщо для зберігання 10-розрядного десяткового числа вимагається 4 байти в пам'яті ЕОМ, а однієї літери або знаку – 1 байт, то щоб запам'ятати і потім відобразити на екрані дисплею кожну точку (так званий піксель – pixel), а їх в залежності від режиму може бути на екрані, наприклад, з адаптером CGA до 163840, необхідно від 1 до 40 біт. При цьому за допомогою одного біта можна закодувати тільки два кольори точки, наприклад пару чорний–білий або синій–голубий, а за допомогою 4 біт – 16 кольорів, вибраних із палітри дисплею. В сучасних комп'ютерах палітра може сягати мільйонів кольорів і відтінків, з яких будь-які 256 можуть одночасно бути присутніми на екрані. Таке розмаїття досягається всього трьома кольорами – червоним (Red), зеленим (Green) і синім (Blue), інтенсивність кожної з яких залежить від яскравості свічення точок люмінофору на екрані, а та, в свою чергу, залежить від інформації, записаної в так званий відеопам'яті, або в відео-RAM дисплею.

Таким чином, якщо умовно прийняти, що для кожної точки екрану потрібно відвести в пам'яті комірку розміром в 3 біти, що відповідають трьом кольорам R, G, B, і записати туди, наприклад, двійкове число 100, то відповідна точка засвітиться червоним (R) кольором; якщо записати в комірку відео-RAM число 001, то відповідна їй точка екрану засвітиться синім (B) кольором, а якщо записати туди 101 пурпуровим (RED+BLUE) кольором. Якщо у всі три біти записати "1", тобто в комірку відео-RAM буде записано число 111, то колір відповідної точки екрану буде білим (R+G+B) у відповідності до закону додавання (змішування) кольорів. (В дійсності, фізичні процеси формування кольору точки на екрані дисплею дещо складніші, оскільки для відображення напівтонів і градацій яскравості вимагається зберігання додаткової інформації, проте для розуміння фізичних основ зберігання графіки у відеопам'яті і відображення її на екрані дисплею досить і цих відомостей).

Пояснивши принцип дії графічного редактора, вчитель повинен запустити MS Paint і дати завдання.

Орієнтований час роботи учнів з графічним редактором – 20-25 хвилин.

Завершивши роботу з програмою, слід провести контрольне опитування.

2. Теоретичні відомості з загальної методики навчання інформатики, що використовуються в лабораторній роботі

Самостійна робота учнів

Самостійна робота – це робота, яка виконується без посередньої участі вчителя за його завданням в спеціально відведений для цього час.

Самостійна робота – це не тільки урок самостійної роботи, але й самостійне вивчення теорії за підручником, самостійне складання алгоритмів, програм, самостійне розв'язування задач, виконання різноманітних завдань тощо.

Самостійна робота є не тільки форма роботи, але й метод навчання.

“Ядром” самостійної роботи є певна, поставлена вчителем, задача.

Структура самостійної роботи – змістовна, процесуальна та мотиваційна.

Ефективність самостійної роботи залежить від:

- 1) змісту матеріалу;
- 2) рівня підготовленості учнів;
- 3) відношення учнів до предмету;
- 4) дидактичних прийомів організації самостійної роботи.

Види самостійної роботи.

1. За глибиною самостійності учнів:

- за зразком;
- реконструктивно-варіативна;
- інтерпретативна;
- творча:
 - розв'язування задач, складання алгоритмів і програм нестандартним або новим для учня способом;
 - розв'язування задач, складання алгоритмів і програм декількома способами;
 - твори з інформатики (наприклад, використання ЕОМ в школі, життя з Internet тощо);
 - доповіді, повідомлення учнів.

2. За глибиною індивідуалізації:

- колективна;
- групова;
- індивідуальна.

3. За дидактичними цілями:

- вивчення нового матеріалу;
- систематизація та узагальнення вивченого;
- закріплення вивченого матеріалу тощо.

4. За джерелом отримання знань:

- робота з підручником;
 - робота з означеннями;
 - переказ прочитаного за планом;
 - відповіді на питання;
 - читання тексту з виділенням головного;
 - читання тексту і складання плану;
 - складання таблиць, алгоритмів, програм на основі прочитаного матеріалу;
- робота з довідковою літературою;

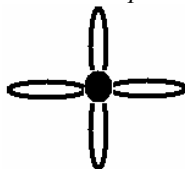
- розв'язування та складання задач, алгоритмів, програм;
 - робота з навчальними вправами;
 - твори з інформатики тощо.
5. *За формою проведення:*
- усні, письмові;
 - домашні, класні;
 - колективні, групові, індивідуальні.
6. *За метою використання самотійної роботи:*
- з метою формування поняття;
 - підготовчі вправи до формування поняття;
 - вправи і задачі на закріплення;
 - вправи і задачі на розвиток і закріплення навичок;
 - вправи і задачі на обчислення з допомогою комп'ютера, розвиток алгоритмічного мислення тощо.

3. Завдання до лабораторної роботи

1. Повторити теоретичний матеріал з курсу інформатики по графічному редактору MS Paint.
2. Розробити фрагмент конспекту уроку пояснення нового матеріалу з теми "Графічний редактор", врахувавши методичні рекомендації. Конспект повинен містити пояснення щодо таких розділів:
 - Графічний редактор - призначення і основні можливості.
 - Вивчення панелі інструментів - інструментів малювання і розфарбовування.
 - Зменшення і збільшення малюнка, стирання, виправлення, дублювання, копіювання, збереження малюнка.
 - Створення текстової рамки і введення тексту.
 - Робота зі вставками, ефектами, оформлення.
3. Розробити систему індивідуальних тренувально-контролюючих завдань з теми (12 варіантів).

Зразок завдання

1) З допомогою інструмента Овал та засобів дублювання отримати такий візерунок:



- 2) Як намалювати пряму строго горизонтальну лінію, пряму строго вертикальну лінію та пряму лінію під кутом 45° ? Проілюструйте.
 - 3) Якими швидкими клавішами викликати вікно Атрибути для встановлення розмірів робочої області?
4. Оформити звіт по лабораторній роботі.

ЛАБОРАТОРНА РОБОТА №12 (8 год.)

Методика вивчення електронних таблиць

ОСНОВНА МЕТА	Розгляд основних методичних особливостей вивчення електронних таблиць MS Excel в школі.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ	Знати основні принципи, форми, прийоми та засоби вивчення електронних таблиць MS Excel.
ППЗ	Електронні таблиці MS Excel.

1. Теоретичні відомості Електронні таблиці MS Excel

Загальні відомості

Електронні таблиці (ЕТ) – це комп'ютерні програми, призначені для збереження та обробки даних, що подаються у табличному вигляді.

Електронні таблиці використовують для:

- *поточних розрахунків* (ведення бухгалтерського обліку, нарахування заробітної плати, проведення переоцінки товару та продукції, калькуляції виконуваних робіт та послуг тощо);
- *аналізу* (оцінювання результатів діяльності);
- *прогнозування* (моделювання реальних ситуацій з оцінкою майбутніх перспектив, розробка планів виробництва продукції і т.п.).

Одним з найбільш популярних засобів управління електронними таблицями є програма *Microsoft Excel*, розрахована на роботу в операційній системі Windows.

Електронні таблиці складаються зі стовпчиків та рядків. Перетин рядка та стовпчика утворює *комірку* таблиці, до якої записуються дані.

Рядок — це горизонтальна лінія комірок. Рядки нумеруються числами.

Стовпчик — це вертикальна лінія комірок. Стовпчики нумеруються однією або двома латинськими літерами.

Кожна комірка має свою адресу, складену із номерів стовпчика і рядка, на перетині яких вона знаходиться. Наприклад, комірка з адресою **AA34** знаходиться на перетині стовпчика **AA** і рядка **34**.

В комірку електронної таблиці можна записати *слово, речення, число* або *формулу* для виконання обчислень.

Якщо значення в комірці визначається самостійно і не може бути визначено з урахуванням значень інших комірок, то дані, що містить така комірка, називаються *основними*. Досить часто значення окремих комірок визначаються за допомогою обчислень на основі значень інших комірок. Дані таких комірок називаються *похідними*.

Для зберігання нового документу Excel створює файл, який називається *робочою книгою*. Кожному наступному файлу Excel надає за замовчуванням чергове ім'я *Книга 1, Книга 2* і т.д.

Робоча книга може мати від одного до 255 робочих листів. Робочі листи мають назви *Лист 1, Лист 2* і т. д.

Робочий лист може мати до 256 стовпчиків і до 65536 рядків.

Після запуску Excel у головному вікні зверху знаходиться рядок заголовка, під ним рядок меню, далі одна або дві панелі інструментів, під ними рядок для запису формул, праворуч – *Область задач*. Внизу знаходяться ярлики робочих листів і діаграм, ще нижче рядок стану.

Комірка A1 окреслена чорним прямокутником (курсором) і називається *поточною*.

Рядок меню складається з дев'яти меню: *Файл, Правка, Вид, Вставка, Формат, Сервис, Данные, Окно, Справка*.

Робота з командами меню, „швидкими клавішами” та панелями інструментів подібна до роботи у Microsoft Word.

Excel забезпечує такі режими подачі документа на екрані (меню *Вид*):

- *Обычный* – встановлюється за замовчуванням для нових документів і дає на екрані найбільше корисної площі для тексту;
- *Разметка страницы* – встановлюється, коли є потреба подивитися, як електронна таблиця розмістилася на аркушах паперу;
- *Во весь экран* – використовується, коли необхідно якнайбільше місця для тексту, що вводиться.

На екрані можуть знаходитись три курсори:

- таблиці;
- клавіатури;
- миші.

Курсор таблиці має вигляд чорної рамки навколо поточної комірки. Перейти на іншу комірку і зробити її активною можна за допомогою миші, стрілочних або спеціальних клавіш і їх комбінацій:

Tab — перехід на одну комірку праворуч;

Enter — перехід на одну комірку вниз;




Home — повернення на початок поточного рядка;

Ctrl + Home — перехід на початок таблиці у комірку A1;

Ctrl + End — перехід до останньої заповненої комірки.

Курсор клавіатури у вигляді I-подібного штриха з'являється в поточній комірці з введенням першого символу тексту, після натискання клавіші F2 або після подвійного клацання лівою кнопкою миші.

Курсор миші може переміщатися по всьому екрану, служить для швидкого переміщення в нову комірку і для подачі команд.

На полі таблиці курсор миші може мати вигляд "товстого плюса" – , "худого плюса" – + або "комбінованої стрілки" – , а в меню команд, на панелі інструментів та в інших місцях – "стрілки" – .



Основні операції

Для *перейменування листа* потрібно на його ярлику двічі клацнути мишею і ввести нову назву (до 31 символу) або виконати команду *Переименовать* у контекстному меню (викликається правою кнопкою миші).

Excel автоматично поміщає в кожну робочу книгу три чистих робочих листи. При необхідності цю кількість можна міняти у проміжку від 1 до 255. Щоб *додати* новий *робочий лист*, необхідно на будь-якому ярлику листа викликати контекстне меню і вибрати команду *Добавить*, а потім вибрати *Лист*. Новий лист буде додано зліва від виділеного. Щоб *видалити лист*, треба у контекстному меню вибрати команду *Удалить*.

Для *переміщення* робочого *листа* треба виділити його ярлик і притримувати натиснутою ліву кнопку миші, поки її курсор не набуде вигляду стрілки з листочком паперу. Перемістити цей курсор вздовж ярликів листів в потрібне місце і після появи маленької чорної стрілки відпустити кнопку.

Для *копіювання* робочого *листа* в тій же робочій книзі треба виконати попередні дії з натиснутою клавішею *Ctrl*. Для копіювання робочого листа в іншу робочу книгу треба використовувати буфер обміну.

При створенні електронної таблиці з першого символу, набраного в поточній комірці, в рядку формул з'являються два значки:  (червоний хрест) – *Відміна* та  (зелена галочка) – *Введення*. В самій комірці з'являється курсор клавіатури. Те, що набирається в поточній комірці, з'являється зразу в двох місцях: в самій комірці і в рядку формул. Поки не подана відповідна команда для зберігання, набрані символи в комірці можна редагувати.

Після набору всіх символів слід натиснути клавішу *Enter* або *Tab* на клавіатурі, або кнопку *Введення* в рядку формул, або клацнути мишею на іншій комірці.


Текст, який не поміщається у комірку, буде "вилазити" на сусідню порожню комірку. Якщо комірка не є порожньою, то текст буде обрізано на границі комірки. Щоб його весь побачити, досить зробити комірку активною і текст з'явиться у рядку формул.

Щоб швидко *ввести* в активну комірку *поточну дату*, досить натиснути клавіші *Ctrl* + ; (латинська крапка з комою).

Щоб швидко *ввести* в активну комірку *поточний час*, досить натиснути клавіші *Ctrl* + : (латинська двокрапка).

В одну комірку можна ввести дату і час з пробілом між ними.

Якщо є потреба ввести число, дату або час у вигляді тексту, досить перед ним набрати апостроф.

Для *зміни ширини стовпчика*, необхідно у горизонтальному заголовку таблиці встановити курсор миші на лінію, що розділяє стовпчики. Коли з'явиться стрілка , натиснути ліву кнопку миші і утримуючи кнопку натиснутою, перетягти стрілку вправо для збільшення ширини стовпчика до потрібного розміру. Подібним чином змінюється висота рядка.

У комірці текст автоматично вирівнюється по лівому краю, а числа і дати – по правому. Щоб *змінити форматування* треба скористатися відповідними кнопками на панелі інструментів.

Щоб змінити форматування зразу в усіх комірках стовпчика (рядка), треба зробити їх активними, для чого клацнути у горизонтальному (вертикальному) заголовку таблиці на позначенні стовпчика (рядка), потім клацнути мишею на відповідній кнопці на панелі інструментів.

Якщо є потреба *виправити помилки* у комірках, то потрібно двічі клацнути на потрібній комірці мишею, з'явиться курсор клавіатури, який переводять у потрібне місце і виправляють помилки. Щоб *стерти запис*, треба зробити комірку активною (клацнути на ній мишею) і натиснути клавішу *Backspace*.

Вставити порожній рядок або *стовпчик* можна, якщо помітити рядок (стовпчик), на місце якого необхідно вставити пустий рядок (стовпчик) та виконати команду *Вставка* → *Строки* (*Столбцы*). Щоб вставити кілька рядків (стовпчиків), необхідно попередньо помітити відповідну кількість рядків (стовпчиків). Після виконання команди помічені рядки (стовпчики) зсунуться вниз (вправо), на їх місце вставляються порожні рядки (стовпчики).

Для *видалення рядка* або *стовпчика* необхідно помітити рядок (стовпчик) та виконати команду *Правка* → *Удалить*. Щоб видалити кілька рядків (стовпчиків), необхідно попередньо помітити відповідну кількість рядків (стовпчиків). Помічені рядки (стовпчики) після виконання команди *Удалить* зчезнуть, нижні рядки (стовпчики справа) зсунуться і займуть їхнє місце.

Над вмістом заповненої комірки або послідовності заповнених комірок можна виконувати операції *переміщення*, *копіювання*, *вставки*, *додавання*, *вилучення*, *відновлення*, *очистки*.

Переміщення — вміст комірки переміщається на нове місце і зникає на старому місці.

Копіювання — вміст комірки копіюється на нове місце і залишається на старому місці.

Вставка — на нове місце вставляється вміст комірки, що переміщається (копіюється), всі комірки залишаються на своїх місцях.

Додавання — на нове місце додається вміст комірки, що переміщається (копіюється), сусідні комірки розсуваються.

Вилучення — вміст комірки вилучається, сусідні комірки зсуваються, щоб зайняти місце вилученої комірки.

Очистка — вміст комірки очищається, сусідні комірки залишаються на своїх місцях.

Відновлення — в комірці відновлюється той вміст, що був до вилучення або очистки.


Excel надає п'ять засобів здійснення операцій:

- використання команд меню;

- використання "швидких клавіш";
- робота з кнопками панелі інструментів;
- контекстне меню;
- автоматизоване заповнення комірок.

Користувач на свій смак вибирає один із засобів або їх комбінацію для роботи над вмістом комірок.

Робота проводиться над вмістом активної комірки або послідовністю активних комірок.

Щоб зробити кілька послідовних комірок активними, досить встановити курсор миші у вигляді "товстого плюса"  на першу комірку, натиснувши і притримуючи ліву кнопку миші, перевести курсор у потрібному напрямку до останньої комірки і відпустити кнопку.

Відмічені таким чином комірки стануть у чорному прямокутнику і будуть *активними* для дій над ними. Така послідовність комірок називається *діапазоном комірок*. Діапазон комірок може бути одномірним (горизонтальним або вертикальним), двомірним, або довільним. Щоб відмітити довільний діапазон, необхідно виділити першу комірку, натиснувши і притримуючи клавішу *Ctrl*, відмітити наступну комірку або групу комірок і так далі, поки не будуть відмічені всі комірки діапазону.

Щоб виділити один стовпчик або рядок, досить клацнути мишею на його заголовку.

Щоб виділити кілька послідовних стовпчиків або рядків, досить при натиснутій лівій кнопці миші протягти її курсор над відповідними заголовками.

Щоб виділити кілька неперіодичних стовпчиків або рядків, досить клацнути мишею на відповідних заголовках, утримуючи натиснутою клавішу *Ctrl*.


Щоб виділити весь робочий лист, досить клацнути мишею на кнопці виділення всього листа у вигляді сірого прямокутника, що знаходиться зліва вгорі на перехресті заголовків рядків і стовпчиків.


Щоб зняти виділення рядків, стовпчиків або всієї таблиці досить клацнути мишею на будь-якій комірці.

Використання команд меню, "швидких клавіш", робота з кнопками панелі інструментів та контекстним меню не відрізняється від подібної роботи у *Microsoft Word*. Новим є *автоматизоване заповнення комірок*.

Активна комірка в правому нижньому куті має маркер заповнення у вигляді чорного квадратика.

	А	В
1	22,4	
2		

Якщо у виділеній комірці курсор миші, який на полі комірки має вигляд "товстого плюса" – , перевести на цей маркер, він набуде вигляду "худого плюса" – +. Такий курсор використовується для автоматичного заповнення послідовності комірок однаковим вмістом.

Якщо у виділеній комірці курсор миші перевести на її границю, він прийме вигляд комбінованої стрілки – . Цей курсор використовується для переміщення вмісту комірки шляхом *буксирування*.

Якщо при цьому притримувати клавішу *Ctrl*, поруч з курсором з'явиться невеликий "плюс". Такий курсор використовується для копіювання вмісту комірки шляхом *буксирування*.

Щоб *перемістити* вміст комірки або діапазону комірок на нове місце, необхідно виконати такі дії:

- виділити комірку або діапазон комірок;
- встановити курсор миші на границю комірки або діапазону комірок, щоб він став "комбінованою стрілкою";
- натиснувши і не відпускаючи ліву кнопку миші, перетягти комірку або діапазон комірок на нове місце;
- відпустити кнопку миші – виділена область комірок очиститься на старому місці і з'явиться на новому.

Якщо при виконанні попередніх дій притримувати клавішу *Ctrl* (курсор миші матиме вигляд "комбінованої стрілки" з маленьким плюсом), виконується *копіювання* вмісту комірки або діапазону комірок на нове місце, на старому місці вміст комірок залишається.

Переміщення або копіювання даних буксируванням використовують при роботі на видимій частині листа

Щоб у наступні комірки *повторити вміст* попередньої комірки, необхідно виконати такі дії:

- зробити комірку з текстом, що буде повторюватися, активною;
- перевести курсор миші на маркер заповнення, щоб перетворився в "худий плюс";
- натиснувши і утримуючи ліву кнопку миші, перевести курсор у потрібному напрямку до кінцевої комірки;
- відпустити кнопку миші, в усіх відмічених таким чином комірках з'явиться текст першої комірки.

Excel дозволяє виконувати *автозаповнення* комірок. Наприклад: можна автоматично заповнити комірки послідовністю назв днів тижня, місяців року або власним списком. Для цього достатньо у активну комірку записати назву першого елемента списку і далі виконати попередні дії по повторенню вмісту.

Щоб створити власний список автозаповнення, необхідно виконати такі дії:

- виконати команду *Сервіс → Параметри → Списки*;
- у полі *Елементи списку* набрати свій список, натискаючи клавішу *Enter* після кожного елемента списку;
- натиснути кнопку *Добавить*.

Робота з числами дещо відрізняється від роботи з текстами. Щоб у наступні комірки *продовжити нумерацію*, вказану у попередній комірці, необхідно виконати такі дії:

- зробити комірку з числом, що буде продовжуватися у наступні комірки, активною;
- натиснувши клавішу *Ctrl*, перевести курсор миші на маркер заповнення, щоб він перетворився в "худий плюс" з додатковим "плюсом";
- утримуючи клавішу *Ctrl* і ліву кнопку миші, перевести курсор у потрібному напрямку до кінцевої комірки;
- відпустити *спочатку* кнопку миші, потім клавішу *Ctrl* – в усіх відмічених комірках з'явиться послідовність

чисел нумерації, які продовжать номер першої комірки.

При наборі перших символів слова Excel пропонує його *автозавершення*, використовуючи зразок із вже набраних подібних слів в тому самому стовпчику. Досить натиснути клавішу *Enter* і варіант завершення буде записано у комірку. Якщо пропонуваний варіант не підходить, продовжують набирати слово. Автозавершення застосовується тільки для тексту. Щоб відмовитися від послуг автозавершення, треба введення тексту розпочати з апострофа.

Excel, подібно до Word, має функцію *автозаміни*, що працює таким же чином.

Обчислення в електронних таблицях

Програма Excel при виконанні обчислень забезпечує точність до 15 знаків, але при виведенні результатів на екран округляє числа у відповідності з форматом комірки, в якій це число відображається.

Округлюючи число, Excel в подальших обчисленнях враховує істинну, а не округлену величину числа.

Щоб встановити *формат* комірки, необхідно виконати такі дії:

- зробити комірку активною або виділити діапазон комірок;
- виконати команду *Формат → Ячейки ...*, відкриється вікно *Формат ячеек*, у якому клацнути на вкладці *Число*;
- у списку вибрати потрібний формат та клацнути мишею на кнопці *OK*.

Excel дозволяє такі *основні формати* комірок.

Общий – для відображення текстів і чисел довільного типу.

Числовой – найбільш загальний спосіб виведення чисел з вибраною кількістю десяткових знаків.

Денежный – для відображення грошових величин.

Финансовый – грошові величини вирівнюються по роздільнику цілої і дробової частини.

Дата – для відображення дати.

Время – для відображення часу.

Процентный – число в комірці множиться на 100 і виводиться із символом відсотка.

Дробный – виводяться прості дробі.

Экспоненциальный – виводяться числа в експонентній формі.

Текстовый – відображаються і обробляються як текст незалежно від вмісту.

Дополнительный – призначений для роботи з базами даних та списками адрес.

Excel дає можливість виконувати різноманітні обчислення в електронних таблицях над вмістом її комірок.

Обчислення виконуються згідно формул. У формулах використовуються адреси комірок та знаки арифметичних операцій: додавання, віднімання, множення, ділення, піднесення до степеня, обчислення відсотка.

Для запису формул у комірку існують такі можливості:

- безпосередній запис у комірку;
- використання рядка формул;
- використання майстра функцій.

При *безпосередньому запису* у комірку та при використанні *рядка формул* користувач записує формулу і її складові частини на власний розсуд і зі своїми помилками залишається сам на сам.

Майстер функцій дозволяє безпомилково вводити функції та їх аргументи, допомагає користувачу вибрати потрібну функцію для формули та її опис. При цьому система знаходить помилки і допомагає їх виправити.

Для запису в комірку формули, необхідно почати зі знака дорівнює =. Наприклад: $=A1+B1$ – це формула, що до вмісту комірки *A1* додає вміст комірки *B1*, а результат записується у комірку, де міститься ця формула.

Формула повинна записуватися в комірку в один рядок.

Для *копіювання формули* з однієї комірки в інші в одному стовпчику або рядку з відповідною автоматичною заміною номерів рядка або стовпчика, треба:

- встановити курсор миші на маркер заповнення поточної комірки, щоб він набрав вигляду "худий плюс";
- натиснути і, утримуючи ліву кнопку миші, протягнути курсор до останньої комірки стовпчика або рядка.

Для позначення діапазону комірок використовується двокрапка як роздільник. Наприклад, формула $=СУММ(A1:A12)$ підраховує суму вмісту комірок з *A1* до *A12*.

Будь-яка формула негайно перераховується, якщо змінюється хоча б одне із її значень.


При *використанні рядка формул* Excel створює сервісні умови для запису формули в активну комірку. Якщо клацнути на символі = в рядку формул, з'явиться діалогове вікно для введення формули.

Формула одночасно з'являється у поточній комірці і у рядку формул. Значення результату показується над формулою. До клацання на кнопці *OK* формулу можна редагувати або відмінити клацанням на кнопці *Отмена*. Після натискання на кнопку *OK* у виділеній комірці з'явиться результат, у рядку формул залишиться відповідна формула, за якою було виконано обчислення.

Excel пропонує більше 200 математичних, фінансових, статистичних і логічних функцій, які немає потреби пам'ятати. Існують засоби для введення функцій і засоби зручного користування ними. Їх завдання – запобігти найбільш типовим помилкам, давати по ходу запису формули підказки, обчислювати проміжні результати.

Функція складається з двох частин: імені і аргументів. Є функції, які не потребують аргументів. Аргументами у функціях найчастіше служать адреса комірок або діапазонів комірок.

Щоб скористатися послугами Excel для введення необхідної функції у формулу, треба виконати такі дії:

- у рядку формул клацнути на символі =;
- ліворуч від рядка формул клацнути на кнопці , відкриється список функцій;



- вибрати потрібну функцію;
- якщо потрібної функції нема у цьому списку, відкрити додатковий список *Другие функции ...*;
- після вибору функції, клацнути на ній мишею, відкриється діалогове вікно для встановлення аргументів (чисел і адрес комірок);
- при встановленні аргументів показується результати обчислення, які можна відмінити і ввести нові, клацнувши на кнопці *Отмена*;
- якщо результат викликає довіру, клацнути на кнопці *OK* і він запишеться у активну комірку.

При копіюванні формули в іншу комірку автоматично міняється адреса комірок, що входять в неї. Це не завжди зручно та потрібно. Адреси комірок можуть бути *абсолютними* або *відносними*.


Відносне посилання у формулі, наприклад *A1*, полягає у відносній позиції комірки, що містить формулу, і комірки, на яку вказує посилання. При зміні позиції комірки, що містить формулу, змінюється і посилання. При копіюванні формули уздовж рядків і уздовж стовпців посилання автоматично коректується. За замовчуванням в нових формулах використовуються відносні посилання.

Абсолютне посилання комірки у формулі, наприклад *\$A\$1*, завжди посилається на комірку, що розташована у визначеному місці. При зміні позиції комірки, що містить формулу, абсолютне посилання не змінюється. При копіюванні формули уздовж рядків і уздовж стовпців абсолютне посилання не коректується.


При необхідності можна зробити абсолютним тільки номер стовпчика (*\$C2*) або рядка (*C\$2*).

Абсолютна адресация комірок досягається в *іменованих* комірках. Комірка або діапазон комірок може мати ім'я.

Застосування імен комірок спрощує побудову та розуміння формул. До цих комірок можна звертатися і з інших листів, тому імена комірок в одній книзі не повинні повторюватися.

Щоб швидко перейти у потрібну комірку, досить клацнути на значку  ліворуч від рядка формул, відкриється список поіменованих комірок, у якому треба вибрати потрібну.

Щоб присвоїти вибраній комірці або діапазону комірок ім'я, треба виділити комірку або діапазон комірок, записати потрібне ім'я в поле адреси активної комірки, що знаходиться ліворуч від рядка формул та натиснути клавішу *Enter*.

Для виклику *майстра функцій* необхідно використати кнопку  у рядку формул.

Відкриється вікно, де треба вказати категорію та вибрати потрібну функцію. Нижче подивитися коротке пояснення про застосування вибраної функції. Клацнути на кнопці *OK*, відкриється вікно *добудовника формул* для встановлення аргументів (чисел і адрес комірок). Вибрана функція запишеться у верхньому віконці добудовника формул разом з необхідними розділовими знаками.

Діалогове вікно добудовника формул допомагає без помилок дописати у формулу потрібні аргументи. Послідовність дій така:

- у віконцях записати аргументи, які автоматично розмістяться у формулу з потрібними розділовими знаками;
- при записі аргументів показуються результати проміжних і загальних обчислень, які можна відмінити і ввести нові, клацнувши на кнопці *Отмена*;
- якщо результат викликає довіру, клацніть на кнопці *OK*, і він запишеться у активну комірку.

При введенні формул Excel здійснює їх перевірку і редагування, використовуючи можливості режиму *Автозамена*. Якщо буде знайдено помилку, то Excel запропонує зробити виправлення за допомогою коректора формул або буде виведено одне з таких повідомлень:

#ДЕЛ/0! – спроба поділити на нуль;

#Н/Д – відповідь неможлива через неповноту даних;

#ИМЯ? – незрозуміла формула, помилка в назві функції або адресі комірки;

#ПУСТО! – незрозуміла адреса комірки, можливо, пропущена кома;

#ЧИСЛО! – проблема з числом у формулі, можливо, загублено аргумент;

#ССЫЛКА! – можливо, стерта комірка, на яку посилається формула;

#ЗНАЧ! – можливо текст замість числа або пуста комірка;

– число не поміщається у комірку по довжині, треба комірку зробити ширшою.

Форматування електронних таблиць

Для документу електронної таблиці виконується:

1. Форматування комірок.
 - 1) встановлення формату чисел;
 - 2) вирівнювання вмісту комірок;
 - 3) зміна виду, розміру, кольору і стилю шрифту;
 - 4) встановлення границь однієї або діапазону комірок;
 - 5) виділення кольором однієї або діапазону комірок;
 - 6) встановлення захисту від випадкового пошкодження або зміни вмісту таблиці;
 - 7) встановлення стилів.
2. Форматування рядків та стовпчиків.
 - 1) зміна висоти рядків і ширини стовпчиків;
 - 2) розміщення тексту в робочому листі.
 - 3) сортування рядків;

- 4) фільтрація даних.
3. Форматування робочих листів.
 - 1) закріплення заголовків робочих листів;
 - 2) розбивка вікна робочого листа на області;
 - 3) групування робочих листів;
 - 4) розміщення робочих листів в окремих вікнах.
4. Форматування книги.
 - 1) створення колонтитулів;
 - 2) встановлення розривів аркушів;
 - 3) попередній перегляд перед друкуванням;
 - 4) друкування електронних таблиць.

Форматування комірок

Щоб встановити потрібний формат треба виконати команду *Формат* → *Ячейки* З'явиться діалогове вікно *Формат ячеек*, в якому треба вибрати потрібну вкладку: *Число*, *Выравнивание*, *Шрифт*, *Граница*, *Вид* або *Защита*.

Число – встановлюють формат виділених комірок.

Выравнивание – встановлюють горизонтальне і вертикальне вирівнювання вмісту комірок, відступ тексту від лівого краю, орієнтацію тексту, розміщення тексту в комірці у кілька рядків, автопідбір ширини, об'єднання комірок.


Шрифт – встановлюють вид, написання, розмір, колір шрифту, а також спеціальні ефекти в написанні тексту.

Граница – встановлюють зовнішні і внутрішні границі навколо і з визначених сторін для виділених комірок у вигляді різного типу, товщини і кольору ліній.

Вид – встановлюють колір та візерунок фону заливки комірок.

Защита – встановлюють режим захисту вмісту комірок: заборона редагування, копіювання та стирання комірок⁹; заборона показу формул в активних комірках.

З однієї комірки в інші можна *копіювати формати*, для чого потрібно виконати такі дії:

- виділити комірку, формат якої буде використано як взірець;
- клацнути мишею на кнопці *Формат по образцу* , що на стандартній панелі інструментів;
- виділити комірки, для яких треба застосувати вибране форматування. Вони наберуть вигляду взірця;
- для багаторазового копіювання формату двічі клацнути мишею на кнопці *Формат по образцу*;
- щоб зняти копіювання формату, слід натиснути клавішу *Esc* або ще раз клацнути мишею на кнопці *Формат по образцу*.

Для збереження встановленого форматування одних комірок з метою використання іншими комірками, потрібно створити *стилі*. Для цього необхідно виконати такі дії:

- встановити в комірці потрібні параметри та виділити її;
- виконати команду *Формат* → *Стиль*, відкриється діалогове вікно *Стиль*;
- набрати ім'я стилю у вікні *Имя стиля* і натиснути *ОК*.

Коли є набір потрібних стилів, форматування комірок стає приємною і необтяжливою процедурою.

Щоб скористатися стилем для форматування комірки або діапазону комірок, треба виконати такі дії:

- виділити комірку або діапазон комірок;
- виконати команду *Формат* → *Стиль*, відкриється діалогове вікно *Стиль*;
- у вікні *Имя стиля* у списку стилів вибрати потрібний стиль, подивитися його характеристики у вікні *Стиль включает*: та натиснути на кнопку *ОК*.

Форматування рядків та стовпчиків

У новій таблиці всі рядки мають однакову висоту, всі стовпчики мають однакову ширину. Під час роботи рядки автоматично збільшують або зменшують свою висоту при зміні розміру шрифту тексту в усіх комірках рядка. Стовпчик, у комірку якого вводиться велике число, автоматично збільшує свою ширину. Але при введенні тексту ширина стовпчика, як правило, не міняється.

Для *автоматичної зміни* розмірів рядків і стовпчиків відповідно до довжини тексту, що вводиться, потрібно виконати такі дії:

- виділити діапазон комірок;
- для рядків вибрати команду *Формат* → *Строка* → *Автоподбор высоты*;
- для стовпчиків вибрати команду *Формат* → *Столбец* → *Автоподбор ширины*.

Для автоматичного настроювання ширини стовпчика або висоти рядка відповідно до найширшого або найвищого запису треба двічі клацнути лівою кнопкою миші на правій границі заголовку стовпчика або нижній границі заголовку рядка.

Excel дозволяє здійснювати *сортування* вмісту комірок. Можна проводити сортування як чисел, так і текстів. Сортування може бути як за зростанням чисел або за алфавітом, так і за зменшенням чисел або проти алфавіту. Можна сортувати всю таблицю або її вказану частину.

Для сортування вмісту стовпчика необхідно виділити стовпчик або його частину, а потім на панелі інструментів натиснути одну із кнопок: *По возрастанию* або *По убыванию*.

Сортування відбувається відповідним переміщенням вмісту комірок у виділеному стовпчику. Вміст інших комірок

⁹ Захист комірок діє лише після захисту всього робочого листа. Щоб захистити лист, треба виконати команду *Защита* в меню *Сервис*, після чого виконати команду *Защитить лист*.

залишається на старому місці.

Щоб *сортувати всю таблицю*, необхідно виконати такі дії:

- виділити всю таблицю;
- виконати команду *Данные → Сортировка...*;
- у вікні *Сортировка диапазона*, що з'явиться, відкрити список *Сортировать по* і вибрати стовпчик, за значеннями якого проводиться сортування;
- вибрати подальші параметри сортування у списках *Затем по* та *В последнюю очередь, по*;
- вибрати вид сортування, клацнувши мишею у полях *по возрастанию* або *по убыванию* та натиснути на кнопки *ОК*.

Сортування відбувається переміщенням рядків у відповідності до вмісту вказаних стовпчиків.

Досить часто виникає потреба вибору із електронної таблиці лише тих даних, значення яких цікавить користувача.

Така процедура називається *фільтрацією* даних. Для фільтрації треба:

- виділити всю таблицю;
- виконати команду *Данные → Фильтр → Автофильтр*, після чого в кожній комірці верхнього рядка з'явиться кнопка для відкриття списку;
- відкрити список, у якому вибрати (*условие...*), з'явиться вікно *Пользовательский автофильтр*;
- задати умови відбору та натиснути кнопку *ОК*.

Форматування робочих листів

Коли таблиця стає великою і не поміщається на екрані, то при просуванні вниз верхні рядки зникають, щоб звільнити місце для нижніх. Точно так при просуванні по таблиці вправо зникають з екрану стовпчики, що знаходяться зліва. Це може призвести до помилок при подальшому заповненні комірок.

Можна *закріпити заголовки* рядків і стовпчиків, щоб вони залишалися на екрані, а інші рядки і стовпчики пересувалися відносно них. Заголовками можуть служити один або кілька рядків і (або) один або кілька стовпчиків.

Щоб закріпити заголовки, треба виділити комірку, що знаходиться нижче від рядка (рядків) і справа від стовпчика (стовпчиків), які треба закріпити як заголовки, а далі у меню *Окно* виконати команду *Закрепить области*. Щоб відмінити цю функцію слід виконати команду *Снять закрепление областей*.

Щоб можна було бачити поруч різні частини робочого листа треба розділити його на дві або чотири області і переглядати кожну окремо. Для цього необхідно виконати такі дії:

- для розбивки вікна робочого листа на верхню і нижню області, треба захопити *віху розбивки* (сірий прямокутник на вершині вертикальної смуги прокрутки) і з появою двонаправленої стрілки перетягти її вниз;
- для розбивки вікна робочого листа на ліву і праву області, треба захопити *віху розбивки* (сірий прямокутник з правого краю горизонтальної смуги прокрутки) і з появою двонаправленої стрілки перетягти її вліво.

Одержані області візуально відокремлені одна від другої *смугами розбивки*, які можна пересувати для збільшення або зменшення тієї чи іншої області. Кожна із областей має свої смуги прокрутки, щоб автономно пересуватися по рядках і стовпчиках. Для відміни розбивки робочих листів на області, досить відповідні віхи повернути на своє попереднє місце.

Якщо в одній книзі створюються кілька подібних робочих листів, краще об'єднати їх у групу. В цьому випадку внесення даних або форматування в одному листі розповсюджується на всі листи групи. В групу можуть входити робочі листи тільки однієї книги.

Щоб *виділити групу* робочих листів, треба по черзі клацнути на їх ярликах при натиснутій клавіші *Ctrl*. Щоб розділити групу, досить при натиснутій клавіші *Shift* клацнути на ярлику першого у групі робочого листа.

Часто необхідне розміщення тексту на кількох *об'єднаних суміжних комірках*. Для цього слід виділити комірки рядка, в які буде розміщено текст та на панелі інструментів клацнути мишею на кнопці *Объединить и разместить в центре*



Інформацію, що міститься в таблиці, можна зобразити у графічному вигляді. Excel пропонує велику кількість діаграм і графіків. Основою всіх цих різноманітних діаграм і графіків є три різновиди: кругова діаграма, стовпчикова діаграма та лінійний графік.

Кругові діаграми більш наочно показують співвідношення частин у цілому. *Стовпчикові діаграми* краще показують кількісні характеристики одночасно в кількох рядках і стовпчиках. *Лінійний графік* краще використовувати для зображення змін показників на протязі визначеного часу.

Створюючи діаграму, Excel зберігає зв'язки між вибраними числовими даними і їх графічним зображенням. Зміни чисел або тексту у таблиці будуть приводити до відповідних змін стовпчиків, секторів та інших елементів діаграм.

Для створення діаграм використовується *Мастер диаграмм*, який основну роботу бере на себе і користувачу залишається тільки робити вибір із його пропозицій. Діаграму можна створити у 4 кроки на окремому робочому листі або вставити в уже існуючий.

Щоб створити діаграму, треба виділити на робочому листі дані, які необхідно відобразити на діаграмі та викликати *Мастер диаграмм* (*Вставка → Диаграмма...*).

Щоб *форматувати* заголовки і написи діаграми, слід виконати такі дії:

- клацнути правою кнопкою миші на заголовку або напису і вибрати команду *Формат* із меню, що з'явиться;
- у вікні *Формат* клацнути на вкладці *Выравнивание* і вибрати потрібні варіанти із списків;
- мишею встановити положення напису у вікні *Ориентация* або ввести точне значення у градусах.

Щоб змінити параметри осей координат треба:

- клацнути правою кнопкою миші на осі, формат якої треба змінити, і вибрати команду *Формат оси* із меню;
- у вікні *Формат оси* клацнути на вкладці *Вид* і вибрати потрібні параметри;
- вибрати параметри для міток поділок в областях *Основное* та *Промежуточные*;

- в області *Метки делений* задати спосіб розташування підписів у поділок.

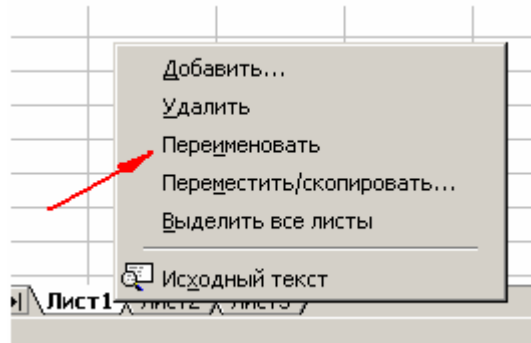
2. Приклад практичної роботи

Практичне завдання №1

Основи роботи з Excel. Введення та форматування даних

Хід роботи:

1. Завантажити програму Microsoft Excel.
2. Ознайомитися з робочим вікном програми. Визначити з кількох робочих листів складається книга.
3. Зберегти робочу книгу у персональну папку.
4. Перейменувати перший робочий лист з імені “Лист 1” на “Штат”
 - *натиснути правою кнопкою миші на зображенні листа,*



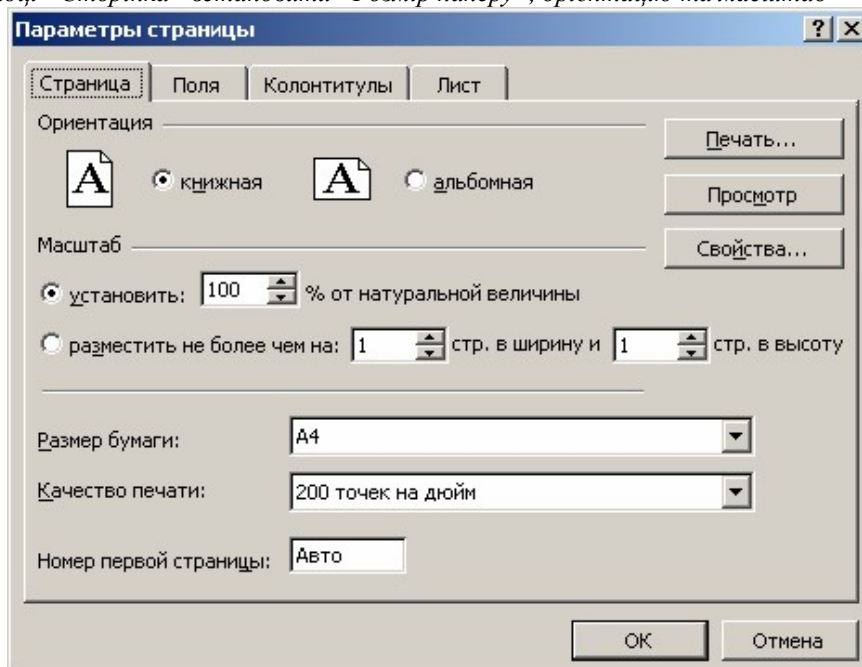
- *знищити старе ім'я листа та написати нове – Штат,*



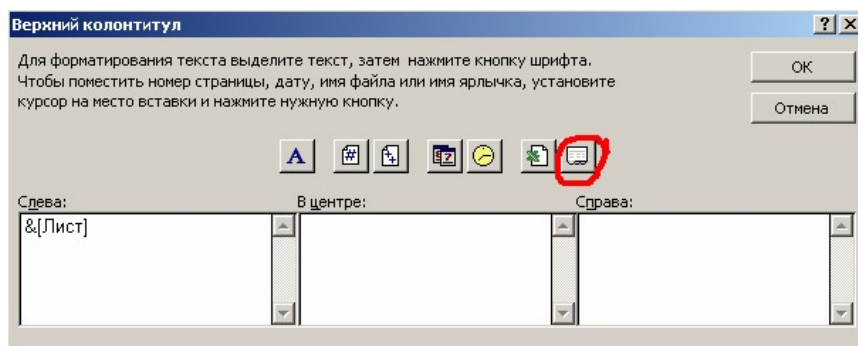
- *натиснути Enter!!!*

5. Встановити наступні параметри сторінки: Формат А4, поля: Ліве – 2 см, Праве – 2 см, Верхнє 1,5 см, Нижнє 1,5 см, Масштаб 90% та створити верхній колонтитул “Назва листа”. Вирівнювання колонтитулу по правому краю.

- *Вибрати пункт меню Файл⇒Параметри сторінки.*
- *На вкладці “Сторінка” встановити “Розмір паперу”, орієнтацію та масштаб*



- *на вкладці поля встановити відповідні розміри полів.*
- *На вкладці колонтитул натиснути кнопку “Створити верхній колонтитул”, а в діалоговому вікні “Верхній колонтитул” встановити курсор в полі “Зліва” та натиснути кнопку “Назва листа”.*



6. В клітинці A1 вписати текст – “Штатний розпис”

- встановити курсор (рамка навколо клітинки) в клітинку A1,
- ввести необхідний текст.

7. Починаючи з клітинки A3 заповнити наступну таблицю

Посада	Кількість	Оклад
директор	1	600
бухгалтер	1	500
продавець	2	450
вантажник	2	350

8. Для стовпчика “Кількість” доцільно використовувати автозаповнення (такий метод застосовується для будь-яких значень, що повторюються):

- встановити курсор на перше значення (B4),
- натиснути ліву кнопку миші на правому нижньому кутку виділення і, не відпускаючи її, протягнути вниз на необхідну кількість клітинок (в нашому прикладі – на одну),
- відпустити ліву кнопку миші.

9. Вставити новий стовпчик перед стовпчиком “Посада” (перед стовпчиком A)

- встановити курсор на будь-яку клітинку стовпчика A,
- вибрати пункт меню Вставка⇒Столбцы.

10. В новостворений стовпець на рівні інших заголовків записати заголовок “№ з/п”.

11. Використовуючи прийом автозаповнення, пронумерувати перелік посад, щоб отримати таблицю наступного вигляду:

№ з/п	Посада	Кількість	Оклад
1	Директор	1	600
2	Бухгалтер	1	500
3	продавець	2	450
4	вантажник	2	350

- в першій клітинці (під клітинкою “№ з/п”) записати перше необхідне значення – “1”,
- виділити цю клітинку,
- натиснути правою кнопкою миші на правому нижньому кутку виділеної клітинки (так як зображено на малюнку), і, утримуючи її натисненою, протягнути вниз на необхідну кількість клітинок (3),

3		
4	№	Посада
5	1	директор
6		бухгалтер
7		продавець
8		вантажник
9		

- відпустити праву кнопку миші,
- в меню, що з’явиться, вибрати пункт “Заполнить”.

12. Клітинку, яка містить назву таблиці – “Штатний розпис” перенести в першу клітинку (з B1 в A1):

- виділити клітинку (B1),
- натиснути лівою кнопкою миші на одній із сторін виділеної клітинки – верхній, нижній, лівий чи правий,
- утримуючи натиснутою ліву кнопку миші, перемістити виділену клітинку в нове місце – куди необхідно перемістити (A1),
- відпустити ліву кнопку миші.

13. Використовуючи стандартні кнопки панелі інструментів, відформатувати вміст таблиці за приведеним нижче зразком:

№	Посада	Кількість	Оклад
1	директор	1	600
2	бухгалтер	1	500
3	продавець	2	450
4	вантажник	2	350

- * для того, щоб застосувати однакове форматування до декількох клітинок, попередньо необхідно ці клітинки виділити,
 * для зміни висоти (ширини) рядка (стовпчика) необхідно підвести вказівник миші до границі назв рядків (стовпчиків) та, утримуючи натиснену ліву кнопку миші, змінити висоту (ширину).

14. Відредагувати вміст клітинок назви таблиці додавши текст “фірми Хлібторг” та клітинки “Оклад” змінивши її на “Посадовий оклад”

Штатний розпис фірми "Хлібторг"						
№	Посада	Кількість	Посадовий оклад			
1	директор	1	600			
2	бухгалтер	1	500			
3	продавець	2	450			
4	вантажник	2	350			

- для того, щоб відредагувати вміст клітинки необхідно виконати одну з наступних дій:

- * або двічі натиснути лівою кнопкою миші на необхідній клітинці,
- * або виділити необхідну клітинку та натиснути F2,
- * або виділити необхідну клітинку та встановити курсор (натиснути) на рядку формул на панелі інструментів
 - виконати необхідні зміни,
 - натиснути Enter!!!

15. Для клітинки “Посадовий оклад” встановити властивість “перенесення по словах”:

- виділити необхідну клітинку,
- вибрати пункт меню *Формат*⇒*Ячейки...*
- вибрати закладку *Выравнивание*,
- встановити перемикач “переносить по словам”
- натиснути на кнопці “Ок”.

16. Для клітинок – заголовків стовпчиків самостійно розглянути можливості форматування на закладці “*Выравнивание*” пункту меню *Формат*⇒*Ячейки...* (а саме можливості вертикального та горизонтального вирівнювання).

17. Додати стовпчик “Сума” та рядок “Всього” так, як зображено нижче:

Штатний розпис фірми "Хлібторг"

№ з/п	Посада	Кількість	Посадовий оклад	Сума
1	Директор	1	600	
2	бухгалтер	1	500	
3	продавець	2	450	
4	вантажник	2	350	
Всього:				

18. Відформатувати клітинки як показано нижче:

Штатний розпис фірми "Хлібторг"

№ з/п	Посада	Кількість	Посадовий оклад	Сума
1	директор	1	600	
2	бухгалтер	1	500	
3	продавець	2	450	
4	вантажник	2	350	
Всього:				

* для зміни типу, стилю та розміру шрифту, а також для зміни кольору шрифту та кольору заливки використовуйте відповідні кнопки на панелі інструментів,

* для виділення групи несуміжних клітин використовуйте ліву кнопку миші при натисненій клавіші CTRL (наприклад для встановлення кольору заливки по стовпчику “Сума” та рядку “Всього”).

19. Розмістити назву таблиці посередині над всіма стовпцями:

- виділити всі клітинки, в яких повинен розміститися текст (в нашому випадку A1...E1),
- натиснути на панелі інструментів кнопку “Объединить и поместить в центре”.

20. Встановити формат клітинок для стовпця “Посадовий оклад” таким, щоб відображалось два знаки після коми (копійки).

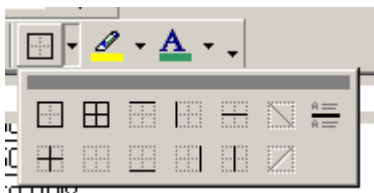
Посадовий оклад
600,00
500,00
450,00
350,00

- виділити діапазон клітинок (всі значення окладів),
- вибрати **Формат**⇒**Ячейки...**,
- вибрати закладку “Число”,
- зі списку числових форматів вибрати “Числовой”,
- встановити “Число десятинних знаків” в значення 2,
- встановити перемикач “Разделитель групп разрядов”,
- натиснути “Ок”.

21. Самостійно переглянути можливі варіанти формату даних з параметрами для клітинок (Денежный, Дата, Время, Процентный та ін).

22. Встановити лінії-роздільники для клітинок таблиці:

- виділити необхідний діапазон клітинок,
- на панелі інструментів вибрати кнопку, що вказана нижче,



- вибрати на панелі, що з'явилася необхідне виділення лініями раніше виділеної групи клітинок.

Практичне завдання №2

Використання формул та функцій в розрахунках. Залежності

Хід роботи:

23. Відкрити з персональної папки робочий файл.

24. На листі “Штат”, де розміщена раніше створена таблиця, необхідно заповнити останній стовпець “Сума”. Значення у кожній з клітинок цього стовпчика – це результат перемноження кількості на посадовий оклад відповідних клітинок кожного рядка. Щоб вставити формулу для такого обчислення, необхідно:

- виділити клітинку (E4), де повинен розміститися результат формули (множення);
- на клавіатурі натиснути кнопку “=” та набрати такий текст C4*D4;
- натиснути Enter;
- в поточній клітинці з'явиться результат множення (при цьому в рядку формул відображується сама формула!).

25. Для наступної клітинки – E5 вставте формулу за допомогою миші (хоча можна скористатися методом, який наведений вище):

- виділити клітинку (E5), де повинен розміститися результат формули (множення);
- на клавіатурі натиснути кнопку “=”;
- при допомозі стрілок управління курсором (←, ↑, →, ↓) або при допомозі лівої кнопки миші вказати першу клітинку даних (перший множник) – виділена клітинка буде відображатися рамочкою, яка переливається);
- потім необхідно на клавіатурі натиснути знак дії (в нашому випадку знак множення, тобто *);
- потім аналогічно вказати другу клітинку даних (другий множник);
- натиснути Enter, що означає кінець введення формули.

26. Надалі, для інших клітинок, що залишилися, можна вводити формули одним із цих двох методів. Але доцільніше скористуватися автозаповненням.

- виділити першу клітинку (E5);
- натиснути в правому нижньому кутку виділення ліву кнопку мишки і не відпускати її;
- перетягнути мишу на необхідну кількість клітинок вниз (в даному випадку на дві);
- відпустити ліву кнопку миші – формули автоматично проставилися в клітинках.

* перегляньте формули для клітинок, побудованих автозаповненням при допомозі рядка формул.

27. Використовуючи один з відомих методів створення формул обрахуйте значення “Всього” для стовпчика “Кількість” (клітинка C9)

* рядок формул повинен містити наступне: =C4+C5+C6+C7

28. Відредагуйте щойно створену формулу, замінивши її на функцію суми:

- перейти в режим редагування необхідної клітинки (C9) (див практичне завдання №1 п.12);
- змінити текст в рядку формул на наступний: =СУММ(C4:C8);

* в дужках вказується діапазон даних, які необхідно просумувати

- натиснути Enter.

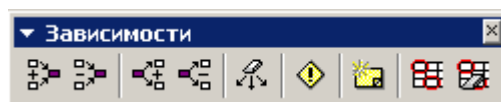
29. Для рядка “Всього” стовпчика “Сума” підрахувати загальну суму, використовуючи при цьому функцію Автосума.

- виділити клітинку (E9), де повинен розміститися результат суми;
- натиснути на панелі інструментів кнопку “Автосума” – Σ;
- група клітинок від E4 по E8 буде відображатися рамочкою, яка переливається;
- при допомозі клавіш управління курсором (←, ↑, →, ↓) або при допомозі лівої кнопки миші вказати необхідний нам діапазон (якщо запропонований автоматично діапазон не задовольняє); тобто лівою кнопкою миші (тримаючи її) провести від першої клітинки до останньої, тобто від E4 по E7; відпустити ліву кнопку миші;
- натиснути Enter.

30. Для стовпчика “Сума” встановити формат “Денежный” з двома десятковими знаками (копійки) та з позначенням “грн.”. Аналогічно зробити для стовпчика “Посадовий оклад”.

31. Розглянути залежності, по яким проводився розрахунок за допомогою інструменту “Зависимости”:

- викликати панель “Зависимости” за допомогою команди “Сервис ⇒ Зависимости ⇒ Панель зависимостей”;



- вибрати клітинку для проведення аналізу (у нашому випадку клітинку суми для рядка “директор”);
- по черговим натисненням кнопок на панелі проглянути можливі залежності;

№ з/п	Посада	Кількість	Посадовий оклад	Сума
1	директор	1	600,00	600,00
2	бухгалтер	1	500,00	500,00
3	продавець	2	450,00	900,00
4	вантажник	2	350,00	700,00
	Всього:	6		2 700,00

- вимкнути панель.

32. Змінити початкові дані в стовпчиках “Кількість” та “Посадовий оклад”. Звернути увагу на те, як будуть змінюватися дані у стовпчику “Сума” та рядку “Всього”.

33. За допомогою кнопки “Предварительный просмотр” (або команди меню “Файл ⇒ Предварительный просмотр”) переглянути вигляд таблиці, підготовленої до друку.

34. Для клітинки “Посадовий оклад” створити примітку:

- виділити клітинку;
- вибрати команду “Вставка ⇒ Примечание”;
- у примітці, яка з’явиться, ввести текст примітки: “Моя примітка” (при необхідності змінити розміри);
- вийти з режиму створення примітки, натиснувши на довільній клітинці;
- як результат створення примітки, клітинка “Посадовий оклад” у верхньому правому кутку матиме мітку червоного кольору, яка вказує на наявність примітки. При підведенні вказівника миші до даної клітинки примітка буде з’являтися автоматично, а при відведенні – зникати;

35. Для редагування примітки слід перейти у режим відображення приміток:

- вибрати команду “Вид ⇒ Примечания”;
- як результат будуть відображені всі примітки і з’явиться можливість їх редагувати, окрім того з’явиться панель “Рецензирование”, яка використовується для створення нових приміток, переходів між примітками і т.п.;

- для примітки клітинки “Посадовий оклад” змінити текст на “При плануванні у аналізі діяльності сетку должностных окладов можно изменять с помощью набора с ц е н а р и е в”;

* видалення приміток також проводиться в даному режимі.

36. Після редагування приміток необхідно повернутися до звичайного режиму відображення таблиці:

- вибрати команду “Вид ⇒ Примечания”.

Практичне завдання №3

Формули та функції. Типи адресації. Автозаповнення. Зв'язок між листами робочої книги

Хід роботи:

37. Відкрити з персональної папки робочий файл.

38. Доповнити штатний розпис іншими посадами згідно наступної таблиці:

- для вставки нових посад використовувати “Вставка⇒Строки”;
- для нумерації використовувати автозаповнення;
- для стовпчика сума використовувати авто заповнення;
- перевірити правильність формул та форматів даних.

Штатний розпис фірми “Хлібторг”

№ з/п	Посада	Кількість	Посадовий оклад	Сума
1	директор	1	600,00 грн.	600,00 грн.
2	бухгалтер	1	500,00 грн.	500,00 грн.
3	продавець	2	450,00 грн.	900,00 грн.
4	вантажник	2	350,00 грн.	700,00 грн.
5	зам.директора	1	500,00 грн.	500,00 грн.
6	касир	1	350,00 грн.	350,00 грн.
7	водій	2	300,00 грн.	600,00 грн.
8	прибиральник	2	150,00 грн.	300,00 грн.
9	секретар-референт	1	270,00 грн.	270,00 грн.
10	охоронець	3	200,00 грн.	600,00 грн.
Всього:		16		5 320,00 грн.

39. Перейти на Лист 2 та перейменувати його на “Кадри”.

40. Використовуючи відомі методи вводу та форматування даних, створити таблицю кадрового забезпечення:

Кадрове забезпечення

№ з/п	ПІБ	Посада	Дата народження	Дата зарахування
1	Головатчук Г.О.			
2	Віцин В.І.			
3	Валютенко В.А.			
4	Банковець Б.А.			
5	Клавіатуренко К.Л.			

41. Заповнити стовпчик “Посада” використовуючи дані таблиці “Штатний розпис” (лист “Штат”):

- виділяємо клітинку посади для першого прізвища в таблиці (C4);
- натиснути знак “=” на клавіатурі або на панелі біля рядка формул;
- при допомозі миші перейти на лист “Штат”;
- виділити клітинку з вмістом “директор” (B4);
- натиснути Enter;
- пересвідчитись, що навпроти першого прізвища з'явилась посада – директор.

42. Аналогічно заповнити весь стовпчик “Посада” згідно зразка:

Кадрове забезпечення

№ з/п	ПІБ	Посада	Дата народження	Дата зарахування
1	Головатчук Г.О.	директор	01.02.80	1 Лют 95
2	Віцин В.І.	зам.директора	02.03.49	1 Лют 95
3	Валютенко В.А.	бухгалтер	02.11.74	1 Лют 95
4	Банковець Б.А.	касир	14.08.78	11 Тра 99
5	Клавіатуренко К.Л.	секретар-референт	02.07.80	15 Лют 97


43. Дати, які записані в стовпчиках “Дата народження” та “Дата зарахування” заповнити самостійно (дані можна підібрати довільні). Формати дат встановити такими, якими вони зображені в таблиці, що наведена вище.
44. Перейти на Лист 3 та перейменувати його на “Розрахунок ЗП”.
45. Створити наступну таблицю:

Розрахункова відомість											
заробітної плати за січень 2002 р.											
		Робочих днів у місяці:		23							
№	ПІБ	Посада	Оклад	Відпрацьовано, годин	Нараховано всього	УТРИМАНО				Утримано всього	До видачі
						Прибутковий податок	Пенсійний фонд	Страховання по безробіттю	Соціальне страхування		

46. Заповнити стовпчик “№” при допомозі авто заповнення від 1 до 16.
47. Заповнити стовпчик ПІБ відповідними даними з листа “Кадри”:
- *перше прізвище вивести аналогічно п5. практичного завдання №3 або написати наступну формулу: “=Кадри!B4” (в результаті з’явиться прізвище директора);*
 - *виділити клітинку з прізвищем директора;*
 - *використовуючи прийом автозаповнення, вивести решту прізвищ.*
48. Використовуючи автозаповнення по горизонталі, вивести першу посаду.
49. Використовуючи автозаповнення по вертикалі, вивести решту посад.
50. Заповнити стовпчик “Оклад”, використовуючи “Спеціальную вставку” для кожної з посад на листі “Штат”.
- *перейти на лист “Штат” та виділити оклад директора (D4);*
 - *скопійювати в буфер обміну вміст клітинки;*
 - *перейти на лист “Розрахунок ЗП” та виділити клітинку окладу директора;*
 - *виконати “Правка⇒Спеціальная вставка”*
 - *у вікні, що з’явиться – “Специальная вставка”, натиснути кнопку “Вставить связь”.*
51. Заповнити стовпчик “Відпрацьовано годин”:
- *виділити клітинку цього стовпчика для першого працівника (директора);*
 - *вставити формулу “=8*D4” (D4 містить кількість відпрацьованих днів);*
 - *використовуючи автозаповнення, проставити аналогічні формули для всіх співробітників.*

УВАГА!!!

При автозаповненні отримані неправильні значення як результат відносної адресації (=8*D4). Тому, при авто заповненні потрібно використовувати абсолютну адресацію! При цьому використовується знак \$ для фіксації стовпчика (перед буквою), рядка (перед цифрою) або одночасно стовпчика і рядка.

52. В зв’язку з неправильною адресацією при авто заповненні, змінити формулу обрахунку кількості відпрацьованих годин з “=8*D4” на “=8*D\$4” (або на “=8*\$D\$4”). Для зміни типу адресації виконати наступне:
- *виділити перше значення (для директора);*
 - *встановити курсор в рядок формул (на координатах клітинки);*
 - *натиснути клавішу F4 на клавіатурі для зміни типу адресації;*
 - *натисніть Enter;*
 - *автозаповненням заповнити решту клітинок цього стовпчика.*
53. Заповнити стовпчик “Нараховано всього” використовуючи формулу “=D8*E8/(\$D\$4*8)” (де D8 – оклад першого в списку співробітника (директор); E8 – відпрацьовано годин першим в списку співробітником (директором); D4 – кількість відпрацьованих днів) та прийом автозаповнення.
54. Розрахувати дані стовпчика “Прибутковий податок”, використовуючи наступний умовний розрахунок:
- якщо нараховано менше 200 грн., тоді прибутковий податок = 10% від нарахованої суми;
- якщо ж нараховано більше як 200 грн., то на 200 грн. нараховується 10%, а на решту суми – 20%
- *виділити першу клітинку стовпчика “Прибутковий податок” (для директора);*
 - *натиснути на панелі інструментів кнопку , або вибрати “Вставка⇒Функція”;*

- у вікні “Мастер функций” вибрати категорію “Логические” та функцію “Если”
- натиснути “Ок”;
- у вікні редагування функції вписати необхідні вирази (див. наступний рисунок) відповідно до умовного розрахунку, прийнятого раніше;

ЕСЛИ

Логическое выражение: $F8 < 200$ = ИСТИНА

Значение_если_истина: $F8 * 10\%$ = 0

Значение_если_ложь: $20 + (F8 - 200) * 20\%$ = -20

= 0

Возвращает одно значение, если указанное условие истинно, и другое, если оно ложно.

Значение_если_ложь: значение, которое возвращается, если логическое_выражение имеет значение ЛОЖЬ.

Значение: 0,00

OK Отмена

- натиснути “Ок”.

- Використовуючи формули та автозаповнення розрахувати для кожного з співробітників значення відрахувань у пенсійний фонд (2% від нарахованого всього), страхування по безробіттю (0,5% від нарахованого всього), соціального страхування (0,5% від нарахованого всього).
- Підрахувати значення “Утримано всього”, як суму всіх відрахувань, та “До видачі” як різницю між “Нараховано всього” та “Утримано всього”. Також підвести підсумки по рядку Всього для всіх нарахованих та утриманих сум.
- Прізвище бухгалтера для підпису відомості взяти як посилання на відповідну клітинку листа “Кадри”.

Розрахункова відомість											
заробітної плати за січень 2002 р.											
Робочих днів у місяці:				23							
№	ПІБ	Посада	Оклад	Відпрацьовано, годин	Нараховано всього	УТРИМАНО				Утримано всього	До видачі
						Прибутковий податок	Пенсійний фонд	Страхування по безробіттю	Соціальне страхування		
1	Головатчук Г.О.	директор	600,00	184,00	600,00	105,55	12,00	3,00	3,00	123,55	476,45
2	Віцин В.І.	зам.директора	500,00	184,00	500,00	85,55	10,00	2,50	2,50	100,55	399,45
3	Валютенко В.А.	бухгалтер	500,00	184,00	500,00	85,55	10,00	2,50	2,50	100,55	399,45
4	Банковець Б.А.	касир	350,00	184,00	350,00	55,55	7,00	1,75	1,75	66,05	283,95
5	Клавіатуренко К.П.	секретар-реф.	270,00	184,00	270,00	39,55	5,40	1,35	1,35	47,65	222,35
6	Сидоренко С.Д.	продавець	450,00	184,00	450,00	75,55	9,00	2,25	2,25	89,05	360,95
7	Махлевська М.А.	продавець	450,00	184,00	450,00	75,55	9,00	2,25	2,25	89,05	360,95
8	Баранов Б.А.	водій	300,00	184,00	300,00	45,55	6,00	1,50	1,50	54,55	245,45
9	Колесник К.О.	водій	300,00	184,00	300,00	45,55	6,00	1,50	1,50	54,55	245,45
10	Тагло Т.Я.	вантажник	350,00	184,00	350,00	55,55	7,00	1,75	1,75	66,05	283,95
11	Кантоваленко К.А.	вантажник	350,00	184,00	350,00	55,55	7,00	1,75	1,75	66,05	283,95
12	Вінник В.І.	прибиральник	150,00	184,00	150,00	16,55	3,00	0,75	0,75	21,05	128,95
13	Відров В.В.	прибиральник	150,00	184,00	150,00	16,55	3,00	0,75	0,75	21,05	128,95
14	Собаченко С.О.	охоронець	200,00	184,00	200,00	25,55	4,00	1,00	1,00	31,55	168,45
15	Цепков П.К.	охоронець	200,00	184,00	200,00	25,55	4,00	1,00	1,00	31,55	168,45
16	Гавкало Г.А.	охоронець	200,00	184,00	200,00	25,55	4,00	1,00	1,00	31,55	168,45
Всього:					5 320,00	834,80	106,40	26,60	26,60	994,40	4 325,60
Бухгалтер					Валютенко В.А.						

- Самостійно змінити значення окладів на листі “Штат” та проаналізувати зміни на листі “Розрахунок ЗП”.

Практичне завдання №4

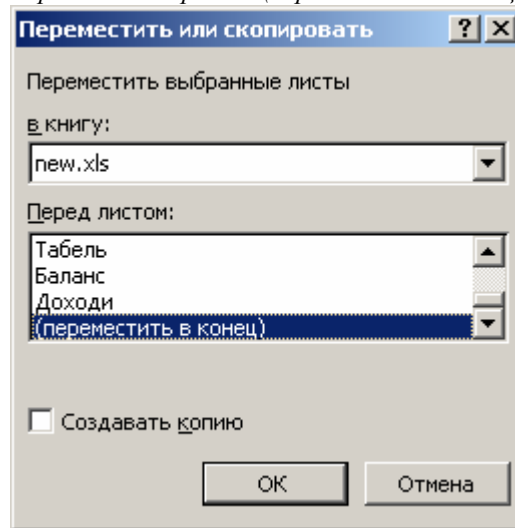
Іменовані клітинки. Діапазони. Списки.

Використання форм, фільтрів, підсумків. Закріплення областей робочого листа

Хід роботи:

- Відкрити з персональної папки робочий файл.
- Додати новий лист:
 - натиснути правою кнопкою миші на будь-якому листі (на останньому);
 - вибрати в меню, що з'явиться пункт “Додати...”;
 - у вікні “Вставка” вибрати “Лист” та натиснути “Ок”;
- Переіменувати новостворений лист на ім'я “Доходи”.
- Перемістити його на останнє місце в списку листів:
 - натиснути правою кнопкою миші на листі, який потрібно перемістити;

- в меню, що з'явилося, вибрати “Переместить/скопировать...”;
- у вікні “Переместить или скопировать” вибрати “(переместить в конец)”



- натиснути “Ок”.

63. На листі “Доходи” створити таблицю за наступним зразком:

	A	B	C	D	E	F	G	H	I	J
	№	Дата	Найменування	Постачальник	Одиниця виміру	Кількість	Ціна закупки	Ціна реалізації	Сума закупки	Сума реалізації (з ПДВ)
1										
2	1	01.11.2001	нарізний	х.з.1	шт	800	0,70			
3	2	01.11.2001	підмосковний	х.з.1	шт	300	0,23			
4	3	01.11.2001	чорний	х.з.1	шт	2000	1,00			
5	4	01.11.2001	формовий	х.з.2	шт	600	0,22			
6	5	02.11.2001	бублик	х.з.1	шт	3000	0,85			

* стовпчик “№” заповніть при допомозі формули, де кожна наступна клітинка в стовпці на 1 більша від попередньої клітинки (для A3: “=A2+1”).

64. Правіше від таблиці (в клітинках L1 та M1 відповідно) записати значення торгової націнки:

Торгова націнка =	10,00%
-------------------	--------

65. Присвоїти ім'я “націнка” клітинці, яка містить значення торгової націнки (M1):

- виділити необхідну клітинку;
- вибрати команду “Вставка ⇒ Имя ⇒ Присвоить ...”;
- у вікні, що з'явиться, у рядку “Имя” ввести нове ім'я клітинки (у нашому випадку “націнка”);
- натиснути “Ок”.

66. Провести розрахунки стовпчиків таблиці:

- “Ціна реалізації”, яка розраховується як сума ціни закупки з націнкою та ПДВ (податку на додану вартість) на цю суму в гривнях: “=(G2+G2*націнка)*120%”;

* націнка – це іменована клітинка, яка створена у п. 7.

- “Сума закупки”, яка дорівнює добутку кількості на ціну закупки: “=F2*G2”;
- “Сума реалізації”, яка дорівнює добутку кількості на ціну реалізації: “=F2*H2”.

67. Використовуючи метод вибору із списку заповнити стовпчики “Найменування” та “Постачальник” до 20 позицій:

- виділити першу вільну клітинку (C7);
- натиснути на ній праву кнопку миші і вибрати із контекстного меню пункт “Выбрать из списка ...”;
- у переліку (побудованому за попередніми значеннями), який з'явиться, вибрати необхідне найменування товару;
- аналогічно провести вибір постачальника;
- розрахункові дані заповнити при допомозі автозаповнення.

68. Доповнити таблицю даними до 55 позицій, використовуючи форму:

- виділіть будь-яку клітинку, яка вже заповнена (входить до списку);
- виберіть пункт меню “Данные⇒Форма...”;

- у вікні “Доходи” додати новий рядок таблиці (причому розрахункові дані заповнюються автоматично)

* використовуючи кнопки «Назад», «Далее» або полосу прокрутки цього вікна, можна переглядати кожний рядок таблиці;

* кнопки «Добавить» та «Удалить» виконують відповідні дії з поточним рядком таблиці.

69. Для того, щоб зручніше переглядати дані великої таблиці, як в нашому випадку, потрібно поставити закріплення областей робочого листа (при використанні скролінгу вмісту таблиці, деяка її частина, наприклад заголовки стовпчиків, будуть залишатися на місці):

- виділити будь-яку клітинку з першого рядка, який повинен піддаватися скролінгу (A2, B2 чи C2 і т.д.);
- вибрати пункт меню “Окно⇒Закрепить области”.

* для зняття закріплення областей – “Окно⇒Снять Закрепление областей”.

70. Відсортувати дані в таблиці по стовпчику “Постачальник” (стовпчик “№” залишити без змін):

- виділити всі дані всіх стовпців, крім першого (“№”), включаючи заголовки стовпців;
- у вікні “Сортировка диапазона” встановити ознаку сортування – по стовпчику “Постачальник” та другорядну ознаку сортування – по стовпчику “Кількість”

- переконатися у виконанні сортування.

71. Аналогічно виконати сортування по стовпчику “Найменування”, “Постачальник”, “Кількість” одночасно.

72. Встановити фільтр для даних таблиці по всіх стовпчиках, крім “№”.

- виділити всі заголовки стовпчиків таблиці (без першого стовпчика);
- вибрати пункт меню “Данные⇒Фильтр⇒Автофильтр”;
- використовуючи створений фільтр, переглянути дані по окремих найменуваннях, постачальниках, кількості і т.д. (як зображено на малюнку нижче, для відбору даних по найменуванню).

	A	B	C	D	E	F	G	H	I	J
	№	Дата	Найменування	Постачальник	Одиниця виміру	Кількість	Ціна закупки	Ціна реалізації	Сума закупки	Сума реалізації (з ПДВ)
1										
2	1	02.11.01	(Все)	х.з.1	шт	3000	0,85	1,12	2 550,00	3 366,00
3	2	05.11.01	(Первые 10...)	х.з.2	шт	1500	0,85	1,12	1 275,00	1 683,00
4	3	25.11.01	(Условие...)	х.з.3	шт	300	0,23	0,30	69,00	91,08
5	4	26.11.01	бублик	х.з.3	шт	1200	0,91	1,20	1 092,00	1 441,44
6	5	02.11.01	нарізний	х.з.3	шт	1300	0,85	1,12	1 105,00	1 458,60
7	6	21.11.01	підмосковний	х.з.4	шт	400	0,70	0,92	280,00	369,60
8	7	23.11.01	рогалик	х.з.4	шт	500	0,70	0,92	350,00	462,00
9	8	28.11.01	формовий	х.з.4	шт	1200	0,85	1,12	1 020,00	1 346,40
10	9	20.11.01	чорний	х.з.4	шт	1500	0,91	1,20	1 365,00	1 801,80
11	10	01.11.01	бублик	х.з.1	шт	800	0,70	0,92	560,00	739,20

73. Зняти авто фільтр:

- *вибрати пункт меню “Данные ⇒ Фильтр ⇒ Автофильтр”.*

74. Встановити проміжні підсумки для кожного з найменувань товару по сумі закупки та сумі реалізації:

- *виділити всю таблицю разом з рядком заголовків стовпців;*
- *вибрати пункт меню “Данные⇒Итоги...”;*
- *у вікні “Промежуточные итоги” вибрати при кожній зміні в – **найменування**, операція – **сума**, додати підсумки по – **сумі закупки та сумі реалізації** (див. малюнок нижче);*

- *натиснути “Ок”;*
- *переглянути проміжні підсумки по кожному з найменуванням та загальний підсумок*

* для згортання/розгортання цілого рівня потрібно натиснути на кнопки відповідного рівня зліва від таблиці (на малюнку позначено цифрою 1);

* для згортання/розгортання підсумку по окремому найменуванню потрібно натиснути на кнопки “+”/“-” (на малюнку позначено цифрою 2).

1	2	3	A	B	C	D	E	F	G	H	I	J
			№	Дата	Найменування	Постачальник	Одиниця виміру	Кількість	Ціна закупки	Ціна реалізації	Сума закупки	Сума реалізації (з ПДВ)
1												
+	11				бублик	Всього					9 106,00	12 019,92
+	26				нарізний	Всього					8 905,00	11 754,60
+	35				підмосковний	Всього					4 918,00	6 491,76
+	46				рогалик	Всього					9 019,00	11 905,08
•	47	42	01.11.01		формовий	х.з.2	шт	600	0,22	0,29	132,00	174,24
•	48	43	31.11.01		формовий	х.з.4	шт	2000	0,85	1,12	1 700,00	2 244,00
•	49				формовий	Всього					1 832,00	2 418,24
•	50	44	22.11.01		чорний	х.з.1	шт	1500	0,91	1,20	1 365,00	1 801,80
•	51	45	01.11.01		чорний	х.з.1	шт	2000	1,00	1,32	2 000,00	2 640,00
•	52	46	21.11.01		чорний	х.з.2	шт	300	0,23	0,30	69,00	91,08
•	53	47	28.11.01		чорний	х.з.2	шт	300	0,23	0,30	69,00	91,08
•	54	48	05.11.01		чорний	х.з.2	шт	900	0,72	0,95	648,00	855,36
•	55	49	03.11.01		чорний	х.з.2	шт	2000	0,85	1,12	1 700,00	2 244,00
•	56	50	27.11.01		чорний	х.з.3	шт	700	0,62	0,82	434,00	572,88
•	57	51	23.11.01		чорний	х.з.4	шт	500	0,85	1,12	425,00	561,00
•	58	52	31.11.01		чорний	х.з.4	шт	750	0,72	0,95	540,00	712,80
•	59	53	29.11.01		чорний	х.з.4	шт	900	0,62	0,82	558,00	736,56
•	60	54	25.11.01		чорний	х.з.4	шт	1500	0,91	1,20	1 365,00	1 801,80
•	61	55	24.11.01		чорний	х.з.4	шт	2000	0,91	1,20	1 820,00	2 402,40
-	62				чорний	Всього					10 993,00	14 510,76
-	63				Общий итог						44 773,00	59 100,36
	64											
	65											

75. Зняти проміжні підсумки:

- встановити курсор на будь-яку клітинку таблиці;
- вибрати пункт меню “Данные⇒Итоги...”;
- у вікні “Промежуточные итоги” натиснути кнопку “Убрать все”;
- натиснути “Ок”.

76. Самостійно встановити підсумки для всіх даних з сумою по стовпчиках: “Кількість”, “Сума реалізації”.

77. Використовуючи аналогічний підхід до створення таблиці, створити на окремому листі “Витрати” таблицю витрат за зразком:

	A	B	C	D	E	F	G
	№	Дата	Вид витрат	Одиниця виміру	Кількість	Ціна одиниці	Сума
1							
2	1	20.11.2001	бензин	літр	20	1,45 грн.	
3	2	20.11.2001	електроенергія	Квт*год	64	0,13 грн.	
4	3	21.11.2001	бензин	літр	10	1,45 грн.	
5	4	21.11.2001	папір друкарський	кг	3	2,75 грн.	
6	5	22.11.2001	стрічка касова	рулон	50	1,75 грн.	
7	6	23.11.2001	ручка кулькова	шт	2	1,25 грн.	
8	7	25.11.2001	бензин	літр	30	1,48 грн.	
9	8	26.11.2001	опалення	Гкал	200	1,75 грн.	
10							

* стовпчик “Сума” розраховувати самостійно

78. Доповнити таблицю “Витрати” на свій вибір різними статтями витрат до 15 рядків.

3 Завдання до лабораторної роботи

1. Повторити теоретичний матеріал з курсу інформатики по електронних таблицях MS Excel.
2. Виконати та проаналізувати практичну роботу, що пропонується учням.
3. В ході аналізу сформулювати всі знання та вміння, що формуються у учнів даною практичною роботою.
4. Розробити фрагменти конспектів уроку пояснення нового матеріалу з теми, враховуючи знання і вміння, якими повинні оволодіти учні при вивченні MS Excel.
5. Розробити систему контрольних питань та завдань для перевірки матеріалу теми.
6. Оформити звіт по лабораторній роботі.

Методика вивчення баз даних

ОСНОВНА МЕТА	Розгляд основних методичних особливостей вивчення СУБД MS Access в школі.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ ПІЗ	Знати основні принципи, форми, прийоми та засоби вивчення СУБД MS Access. СУБД MS Access..

1. Теоретичні відомості

Система управління базами даних MS Access

Бази даних. Система управління базами даних. Структура та початкові відомості про MS Access

База даних (БД) – це будь-яка сукупність зв'язаної інформації, що об'єднана за деякою ознакою. В інформатиці під базою даних здебільшого розуміють будь-який набір взаємозв'язаних структурованих даних, які зберігаються комп'ютером.

Однією із характеристик бази даних є наявність зв'язків між елементами в ній. За способом встановлення таких зв'язків бази даних поділяють на *ієрархічні, мережіві та реляційні*.

Ієрархічна модель будується за принципом підпорядкованості типів об'єктів, тобто встановлюється певний однозначний зв'язок між головним та підлеглими об'єктами.

У *мережній* моделі один і той самий об'єкт може одночасно виступати як в ролі головного, так і підлеглого елемента, тобто кожен об'єкт може брати участь в довільній кількості зв'язків.

У *реляційній* моделі даних об'єкти і взаємозв'язки між ними представляються за допомогою прямокутних таблиць. База даних містить множину таблиць, зв'язок між якими встановлюється за допомогою співпадаючих полів. Комбінація всіх таблиць і їх взаємних зв'язків складає основу бази даних. Завдяки своїй простоті і природності представлення реляційна модель стала найпоширенішою.

Системою управління базами даних (СУБД) називають програмне забезпечення, що реалізує виконання операцій над даними бази.

Головна роль СУБД полягає в забезпеченні користувача інструментарієм, який дозволяє оперувати з даними, що зберігаються в персональному комп'ютері. В СУБД база даних і прикладна програма нерозривно пов'язані одне з одним.

Популярною СУБД є *Microsoft Access* – програма офісного пакету Microsoft Office, у якій передбачені усі необхідні засоби для визначення та обробки даних, а також для керування ними при роботі з великими обсягами інформації.

Під базою даних в Access розуміють сукупність основних об'єктів: *таблиць, запитів, форм, звітів, макросів та модулів*.

Як зазначалось, у реляційних базах даних, до яких відноситься Access, вся інформація міститься у таблицях.

Таблиці – це основні об'єкти бази даних. На їх основі будуються форми, запити та звіти. Кожен рядок таблиці являє собою *запис*, а стовпець – *поле*. База даних може складатися з декількох таблиць. Всього до складу бази даних Access може входити не більше 32768 таблиць, причому одночасно можна відкрити для доступу 255 таблиць.

Запити – це спеціальні структури, що у першу чергу призначені для пошуку та відбору даних за встановленими критеріями.

Форма – це бланк, який потрібно заповнити для доповнення таблиці. Форма дозволяє спростити процес заповнення бази та уникнути помилок, що виникають при безпосередньому введенні інформації до таблиць.

Звіти служать для відображення підсумкових даних з таблиць та запитів у зручному для перегляду та друку вигляді, а також для проведення аналізу даних.

Макроси – це набір команд або дій, які виконує користувач і які автоматизують процеси заповнення бази даних, відбору інформації тощо.

Модулі – це програмні процедури, написані на мові Visual Basic і які реалізують додаткові процедури обробки подій чи виконання обчислень.

Одним із найпотужніших засобів Access є можливість встановлювати *зв'язки* між записами декількох таблиць. Зв'язки між таблицями не є обов'язковими, однак, це гарантує отримання правильних результатів при розробці форм і звітів.

Зв'язок встановлюється між однаковими за типом полями таблиць бази даних, проте не обов'язково, щоб ці поля мали однакову назву.

Для баз даних можна встановити такі типи зв'язку: *один-до-одного, один-до-багатьох, багато-до-багатьох*.

При зв'язку *один-до-одного* кожен запис однієї таблиці зв'язаний лише з одним записом іншої. Найчастіше потреба в цьому виникає, коли треба розбити одну таблицю на декілька дрібніших, оскільки деякі відомості з неї часто не використовуються або деякі дані мають конфіденційний характер.

У більшості випадків будь-які дві таблиці зв'язуються відношенням *один-до-багатьох*. Це означає, що будь-якому запису першої таблиці може відповідати декілька записів другої, однак будь-який запис другої таблиці може зв'язуватись лише з одним записом першої.

При зв'язку *багато-до-багатьох* будь-якому запису першої таблиці може відповідати декілька записів другої та навпаки.

Між двома таблицями може буди заданий лише один зв'язок, тип якого можна, при потребі, змінити.

Програма Access є складовою пакету програм MS Office, тому вікно програми містить усі стандартні елементи вікон програм пакету, таких як Excel чи Word. Однак, є певні особливості.

Після запуску програми з'являється *головне вікно програми*, яке є робочим простором, де розташовуються інші вікна програми. У правій частині знаходиться *Область задач*, на якій користувач може вибрати один з варіантів продовження роботи: створення нової бази за допомогою конструктора чи майстра баз даних або відкриття існуючої бази даних.

Головне вікно бази даних дозволяє отримати доступ до всіх об'єктів бази даних та вибрати режим роботи з ними. Це вікно має свій *рядок заголовку*, в якому відображається ім'я бази даних та кнопки режимів відображення вікна, *панель інструментів*, *панель об'єктів* бази даних та *панель групування*.

Об'єкти, які входять у базу даних, розміщено на відповідних вкладках *панелі об'єктів*. Активізувавши будь-яку з них, користувач відкриває перелік існуючих об'єктів вибраного типу. Якщо вибраних об'єктів немає, то у вікні бази за замовчуванням знаходяться лише ярлики, які дозволяють вибрати один із можливих способів створення того чи іншого об'єкта.

Структура та доступність пунктів меню залежить від режиму роботи Access. Програма має два основні режими роботи:

- *експлуатаційний* режим, в якому відображаються вікна об'єктів бази даних, а також виконуються операції доповнення, вилучення, оновлення, вибір, аналіз даних тощо;

- *проектувальний* режим, в якому можна створювати та модифікувати структури та макети об'єктів бази даних.

Для *запуску експлуатаційного режиму* потрібно виділити об'єкт (таблицю, форму тощо) та натиснути кнопку *Открыть* панелі інструментів вікна об'єктів бази даних або двічі клацнути по потрібному об'єкту лівою кнопкою миші.

Для *запуску проектувального режиму* об'єкт активізують та клацають по кнопці *Конструктор* панелі інструментів вікна об'єктів бази даних.

Програма Access дозволяє створювати бази даних двома способами: за допомогою спеціальної програми – майстра створення бази даних або самостійно створюючи об'єкти бази (таблиці, запити, форми та інші).

Майстер створення бази даних дозволяє на основі вибраного зразка, створити файл, у якому знаходитимуться певні об'єкти – порожні таблиці, форми та звіти.

Створити свою базу даних на основі існуючого шаблону можна наступним чином. На *Области задач* програми вибрати функцію *Общие шаблоны*. Після цього відкриється діалогове вікно, в якому на вкладці *Базы данных* вибрати шаблон, за яким повинна створюватись база даних. Вказівки та запитання майстра допомагають вибрати поля таблиць шаблону БД, структури яких використовуватимуться в новій базі.

Самостійне створення бази даних дозволяє користувачеві самому визначити структуру бази даних та кожного окремого об'єкта, використовуючи шаблони окремих об'єктів, майстри проектування або режим конструктора.

В останньому випадку слід в *Области задач* програми вибрати функцію *Новая база данных*. Після цього відкриється діалогове вікно *Файл новой базы данных*. За замовчуванням програма присвоює кожному новому файлу бази даних ім'я db1, db2 і т.д. У списку *Папка* вибирають папку чи диск, де знаходитиметься файл бази даних, у полі *Имя файла* задають назву файлу та натискають кнопку *Создать*.

Після цього відкриється вікно нової бази даних, в якому можна розпочинати процес створення об'єктів БД.

Таблиці

Першим кроком при створенні бази даних є проектування таблиць.

Кожна таблиця Access має жорстку структуру, тому вся інформація в ній повинна бути чітко систематизована. Для кожної теми в базі даних відводиться окрема таблиця. Це дозволяє уникнути повторень при збереженні інформації та зменшує ймовірність виникнення помилок при введенні інформації.

Назва таблиці не повинна перевищувати 64 символів. В імені таблиці дозволяється використовувати пропуски, але не можна вживати крапку, знак оклику та квадратні дужки.

Кожне поле запису повинно мати назву (ім'я поля) та певні властивості.

Необхідною умовою для нормальної роботи з базою даних є однозначне визначення кожного запису. Елемент запису (поле), по якому можна визначити значення інших елементів даних цього запису, називається *ключовим елементом* даних (*ключовим полем*). У реляційній базі даних кожна таблиця повинна мати ключовий елемент – одне або декілька полів, які однозначно визначають запис таблиці.

За замовчуванням ключові поля у списку даних таблиці відображаються першими. У ролі ключових полів можна використовувати лише дані, які ніколи не будуть повторюватись. Найчастіше ключовими встановлюють номери записів, рахунків, фондів, накладних та ін. Первинним ключем не слід використовувати прізвища, оскільки вони можуть повторюватися.

Дані в таблицях записуються у відповідні поля. Залежно від того, яка інформація буде розміщена в кожному полі, встановлюється відповідний формат поля, який визначається типом даних. Програма Access для збереження та обробки інформації використовує такі *типи даних*:

текстовый – текст, обсяг якого не перевищує 255 символів;

поле MEMO – текст або числова інформація обсягом до 65535 символів;

числовой – числові значення до 16 байтів;

дата/время – значення дати та часу;

денежный – числа, які містять 15 розрядів зліва від десяткової крапки та 4 розряди справа від неї;

счетчик – унікальне значення, яке автоматично змінюється, коли до таблиці додають новий запис;

логический – значення *Так (1)* або *Ні (0)*;

поле объекта OLE – об'єкти, створені та зв'язані з іншими програмами;

гиперссылка – шлях до файла на жорсткому диску або адреса, яка вказує на розташування інформації;

мастер подстановок – створення нових полів з певним списком підстановок.

Створена у базі даних таблиця може відображатися у двох режимах: в *режимі таблиці* та в *режимі конструктора*.

Режим конструктора дозволяє змінювати лише структуру таблиці (назви та властивості полів), а режим таблиці – як структуру, так і вміст даних.

Щоб відобразити таблицю в *режимі таблиці* потрібно двічі клацнути по піктограмі таблиці на вкладці *Таблицы* або виділити піктограму таблиці та натиснути кнопку *Открыть* панелі інструментів вікна об'єктів бази даних.

Щоб відобразити таблицю в *режимі конструктора* потрібно виділити її піктограму та натиснути кнопку *Конструктор* панелі інструментів вікна об'єктів.

Для переходу з одного режиму в інший треба скористатися кнопкою *Вид* панелі інструментів вікна бази даних.

У програмі Access таблицю можна створити одним із способів:

- у режимі таблиці;
- за допомогою майстра таблиць;
- у режимі конструктора;
- шляхом імпорту готової таблиці;
- за допомогою встановлення зв'язку з іншою базою даних.

Вибір певного способу здійснюється при натисканні по кнопці *Создать* вікна об'єктів бази даних. З'явиться вікно *Новая таблица*, де вказані можливі способи створення таблиці – *Режим таблицы*, *Конструктор*, *Мастер таблиц*, *Импорт таблиц*, *Связь с таблицами*.

Створити таблицю можна також зразу ж після відкриття бази даних, вибравши вкладку *Таблицы*, де є три ярлики для створення нової таблиці: *Создание таблицы в режиме конструктора*, *Создание таблицы с помощью мастера*, *Создание таблицы путем ввода данных*.

Режим таблицы

Для створення таблиці у цьому режимі на вкладці *Таблицы* потрібно вибрати ярлик *Создание таблицы путем ввода данных*. З'явиться порожня таблиця, поля якої за замовчуванням мають назви *Поле1*, *Поле2* и т.д. Перейменувати стовпці можна двічі клацнувши по назві стовпця та ввівши нове ім'я, або виділивши поле, вибрати в його контекстному меню команду *Переименовать столбец*.

Для збереження створеної таблиці натискають кнопку закриття вікна таблиці та у діалоговому вікні *Сохранение* вводять назву таблиці.

Мастер таблиц

Майстер таблиці – це спеціальна програма, яка допомагає користувачеві, послідовно виконуючи вказівки, створювати таблицю на основі існуючих шаблонів. Для запуску майстра потрібно на вкладці *Таблицы* активізувати ярлик *Создание таблицы с помощью мастера*. Відкриється вікно *Создание таблиц*, в якому необхідно, вибрати зразок таблиці та поля, які будуть у таблиці (можна вибирати поля з декількох шаблонів).

Для того, щоб назва поля шаблону опинилася в структурі нової таблиці, потрібно двічі клацнути по ній лівою кнопкою миші, або виділити та натиснути кнопку > вікна. Для перенесення всіх полів зразка слід натиснути кнопку >> цього ж діалогового вікна. Забирають поля зі структури кнопками < або << відповідно.

Для зміни назв полів у списку нової таблиці потрібно активізувати поле та натиснути кнопку *Переименовать поле...*

Далі задають ім'я нової таблиці, вибирають один із способів визначення ключового поля, задають при потребі типи зв'язків з уже існуючими таблицями бази, змінюють структуру таблиці, вводять дані тощо. Після натискання кнопки *Готово* на вкладці таблиць з'явиться піктограма нового об'єкту.

Режим конструктора

Режим конструктора дозволяє створювати структуру таблиці самому користувачеві, задаючи тип кожного поля та відповідні властивості. Відобразити вікно конструктора можна з вкладки *Таблицы*, вибравши ярлик *Создание таблицы в режиме конструктора*.

Структура таблиці формується в області проекту, яка складається з трьох стовпців – *Имя поля*, *Тип данных*, *Описание*. Нижня частина вікна конструктора *Свойства поля* стає доступною лише після введення назви поля та вибору його типу.

У рядки стовпця *Имя поля* вводять назви полів майбутньої таблиці. За допомогою клавіші *Enter* або *Tab* переміщують курсор у наступний стовпець *Тип данных*, де за замовчуванням встановлюється тип поля *Текстовый*. Для зміни типу натискають кнопку, яка розгортає список типів полів.

У стовпчик *Описание* можна вводити додаткову інформацію. Це не є обов'язковим, однак такі підказки полегшать в подальшому роботу з базами даних.

Кожне поле проекту таблиці має певну сукупність властивостей, які відображаються в області *Свойства поля*. Активізуючи будь-яке поле у верхній частині конструктора, на вкладці *Общие* області властивостей з'являється сукупність значень параметрів, встановлених за замовчуванням. Користувач при потребі може змінити ці параметри.

Щоб позначити поле як ключове, його активізують та натискають кнопку *Ключевое поле* панелі інструментів або подібну функцію контекстного меню. Поле первинного ключа позначиться спеціальним маркером у вигляді ключа.

Структуру готової таблиці можна змінити, однак лише до наповнення таблиці даними. Інакше це може призвести до втрати даних.


Для збереження структури таблиці потрібно клацнути по кнопці закриття вікна конструктора таблиці і ввести назву таблиці у поле *Имя таблицы* діалогового вікна *Сохранение*.

Введения даних

Заповнення таблиці даними відбувається лише в режимі таблиці.

Для того, щоб в поле ввести інформацію, слід помістити курсор у відповідну комірку і ввести з клавіатури потрібні символи.

Щоб утворити новий запис в таблиці, можна:

- або перемістити курсор в кінець таблиці та натиснути клавішу *Enter*;
- або натиснути кнопку  області *Запись*;
- або виконати команду *Вставка → Новая запись*.

Для *редагування* даних потрібно активізувати відповідний запис. Під час редагування запису зліва від нього з'являється зображення олівця.

Після переміщення курсору на інший запис, зміни автоматично записуються в таблицю.

В Access неможливо редагувати поле типу *Счетчик* – значення в полях лічильника встановлюються і зберігаються автоматично.

Щоб *знищити* запис, потрібно його виділити та:

- або натиснути клавішу *Del* клавіатури;
- або натиснути кнопку *Удалить запись* панелі інструментів;
- або задіяти функцію *Удалить запись* контекстного меню.

В Access процес знищення записів незворотній.

Зв'язки між таблицями

Зв'язки між таблицями відображаються у вікні *Схема данных*, яке викликається командою *Сервис → Схема данных* або натисканням кнопки *Схема данных* панелі інструментів.

Якщо зв'язки між таблицями ще не були встановлені, то автоматично відкриється вікно *Добавление таблицы*. Це вікно можна також викликати (при додаванні нових таблиць) натисканням кнопки *Отобразить таблицу* панелі інструментів *Связь* або виконавши команду *Связи → Добавить таблицу* чи відповідну команду контекстного меню вікна *Схема данных*.

У вікні *Добавление таблицы* потрібно по черзі вибирати зі списку існуючих таблиць ті, які будуть брати участь у зв'язках, та натискати кнопку *Добавить*. У вікні *Схема данных* з'являться вікна з полями зв'язаних таблиць.

Якщо при створенні структури таблиці у полі підстановок були використані значення іншої таблиці, то при виведенні схеми зв'язків будуть відображені відповідні зв'язки (лінії, які з'єднують дві таблиці).

Зв'язок між таблицями можна задати і вручну. Для цього потрібно у вікні *Схема данных* відобразити вікна зв'язаних таблиць та перемістити за допомогою миші (при натиснутій лівій кнопці) поле однієї таблиці до відповідного поля іншої. Після відпускання кнопки з'являється діалогове вікно *Изменение связей*, в якому задають параметри зв'язку.

Задати чи змінити параметри можна і вже встановлених зв'язків. Для цього потрібно активізувати зв'язок (лінію, яка з'єднує дві таблиці) та виконати команду *Связь → Изменить связь* або викликати контекстне меню лінії зв'язку та вибрати аналогічну команду.

У вікні *Изменение связей*, зокрема, встановлюється *Обеспечение целостности данных*, що дозволяє уникнути помилок при знищенні або при доповненні даних у зв'язаній таблиці.

Встановивши зв'язки між таблицями, отримаємо схему даних із позначками **1** та ∞ на кінцях графічних зображень (символ **1** визначає таблицю з боку „один”, а символ ∞ – таблицю з боку „багато”).

Впорядкування таблиць

Для впорядкування записів за даними одного поля встановлюють курсор у це поле та використовують відповідну команду *Записи → Сортировка* або кнопки *Сортировка по возрастанию* чи *Сортировка по убыванию* панелі інструментів.

Для впорядкування за декількома полями (максимум 10 полів), потрібно розмістити поля в потрібній послідовності сортування, виділити ці поля та встановити порядок сортування (за допомогою команди меню чи кнопки панелі інструментів).

Під час роботи з базою даних важливу роль відіграє пошук та аналіз інформації, яка задовольняє певним умовам.

Фільтрація даних – це відображення на екрані лише тих записів таблиці, які задовольняють певним умовам.

Програма Access використовує три види фільтрації:

- *просту фільтрацію* – пошук інформації, яка задовольняє хоча б одній з заданих умов;
- *фільтрацію за зразком* – відбір даних, які співпадають з виділеним фрагментом;
- *розширену фільтрацію* – відбір інформації, яка відповідає складним критеріям пошуку.

Для створення фільтра потрібно відобразити таблицю в режимі таблиці.

Перед створенням будь-якого фільтра, потрібно пересвідчитись у тому, що на дані таблиці не накладено жодних умов, тобто що всі фільтри знищено (при цьому кнопка *Применение фильтра* панелі інструментів буде неактивна).

Простий фільтр

Вікно конструктора простого фільтра відобразиться після виконання команди *Записи → Фильтр → Изменить фильтр* або натискання кнопки *Изменить фильтр*. Після цього відкриється вікно конструктора звичайного фільтра.

На екрані відобразиться рядок таблиці, кожна комірка якого містить список значень відповідного поля. У нижній частині вікна розміщені ярлики *Найти* та *Или*. Вони використовуються для створення складних умов.

Для створення критерію відбору даних потрібно вибрати зі списку поля відповідне значення або ж ввести з клавіатури потрібну умову.

Як зазначалося, звичайний фільтр дозволяє здійснювати фільтрування за складними умовами. Умови, зв'язані логічним оператором *І*, заносяться в один рядок конструктора, а умови, зв'язані оператором *АБО*, заносять в новий рядок конструктора, який відобразиться на екрані після натискання ярлика *Или*. Після формування усіх потрібних умов, натискають кнопку *Применение фильтра* – на екрані відобразяться відібрані записи.

Для відображення усіх захованих записів ще раз натискають кнопку *Применение фильтра*, яка при активному фільтрі змінює назву на *Удалить фильтр*, або виконати команду пункту меню *Записи → Фильтр → Удалить фильтр*.

Для того, щоб зняти усі накладені умови на дані таблиці, потрібно очистити поля конструктора фільтра та запустити фільтр на виконання ще раз.

Фільтр за зразком

Для того, щоб відобразити записи, в яких значення поля співпадають із виділеним фрагментом, використовують фільтрування за зразком. Для цього виділяють потрібний фрагмент поля та виконують команду пункту меню *Записи* → *Фільтр* або натискають кнопку *Фільтр по выделенному*.

Щоб знову відобразити усі записи, натискають кнопку *Удалить фильтр*.

Умови, які формуються при створенні фільтру за виділеним фрагментом, можна переглянути у вікні простого фільтру. Для цього достатньо натиснути на кнопку *Изменить фильтр* – на екрані відобразиться вікно конструктора простого фільтру з занесеною умовою. Якщо при фільтруванні за зразком був фрагмент поля, то умова містить оператор *Like* (подібний до) та символи шаблону *, який вказує на те, що на його місці може бути будь-яка кількість довільних символів. При створенні умов можна також використовувати символ, який замінює один довільний символ.

Для виведення даних, які задовольняють одночасно декілька умов (умови зв'язані логічним *И*), потрібно по черзі використати фільтри за відповідними зразками.

Якщо потрібно вибрати інвертовані дані, то виділяють потрібний фрагмент даних та вибирають команду *Исключить выделенное* або вказують умову з використанням оператора *Not* чи символів <>.

Розширений фільтр

Вікно розширеного фільтра відкривається після виконання команди меню *Записи* → *Фільтр* → *Расширенный фильтр*.

Вікно конструктора розширеного фільтру складається з двох частин: верхньої, в якій розміщено вкладене вікно таблиці зі списком усіх її полів, та нижньої – бланка фільтра, де формують умови відбору та впорядкування.

У вікні конструктора розширеного фільтра можна побачити всі критерії відбору даних, які встановлювались при створенні звичайного фільтра чи фільтра за зразком. Для створення критерію фільтрації, потрібно у поле бланку *Поле* ввести назву поля таблиці, на яке накладатиметься умова. Вказану дію можна зробити:

- або двічі клацнувши по полю у вікні таблиці, яке знаходиться у верхній частині вікна;
- або виділивши поле чи декілька полів за допомогою клавіші *Ctrl* у вікні таблиці та перетягті їх мишею;
- або вибравши потрібне поле зі списку у комірці рядка *Поле*.

Для створення складного критерію у бланк фільтра вносять назви усіх тих полів, на які накладаються умови.

На перетині рядка *Условие отбора* та відповідного поля потрібно ввести критерій. Можна одночасно передбачити сортування відфільтрованих записів, задавши відповідну опцію у рядку *Сортировка*.

Якщо потрібно вивести записи з даними, які задовольняють декільком умовам, об'єднаних логічним *АБО* для одного поля, то у відповідній комірці вводять вирази, об'єднані оператором *Or* або записують окремі умови у рядках одного стовпця (після слова *Или* бланку розширеного фільтра).

У цьому ж вікні можна встановити параметри впорядкування записів за даними декількох полів. Для цього у нижній частині вікна конструктора розширеного фільтра розміщують поля в послідовності, за якою відбуватиметься сортування. Параметри сортування встановлюють у рядку *Сортировка*.

При закритті таблиці зі збереженням змін, незалежно від способу фільтрації чи сортування, зберігається і сам фільтр. Для збереження таблиці без фільтра необхідно відмінити його дію, очистити усі критерії відбору та закрити таблицю зі збереженням змін.

Запити

У СУБД Access для пошуку та відбору інформації використовують запити – об'єкти, які окрім пошуку потрібних даних, дозволяють їх змінювати та аналізувати.

Результатом виконання запитів є тимчасовий масив даних. У такій таблиці можна додавати, змінювати та знищувати записи, проте такі зміни не відображаються на даних основних таблиць. Сам запит можна зберегти як окремий об'єкт, але дані, отримані при його виконанні, зазвичай не зберігаються.

Запити за своїм призначенням та результатами відбору поділяються на:

- запити на вибірку;
- запити на зміну даних (доповнення, знищення, оновлення, створення нової таблиці);
- запити з параметрами;
- перехресні.

Запит можна відкрити в режимі таблиці (запустити запит на виконання), в режимі конструктора або в режимі мови програмування SQL.

Запустити запит на виконання можна у вікні бази даних на вкладці *Запросы*, виділивши потрібний запит, натиснути кнопку *Открыть* панелі інструментів вікна бази даних або двічі клацнувши курсором миші по піктограмі запиту.

Відкрити запит в режимі конструктора можна, виділивши потрібний запит на вкладці *Запросы*, натиснути кнопку *Конструктор* панелі інструментів вікна бази даних.

Перейти з режиму конструктора в режим таблиці можна за допомогою кнопки *Запуск* панелі інструментів конструктора запитів або елемента списку кнопки *Вид*.

Програма Microsoft Access дозволяє створювати запити в режимі конструктора та за допомогою спеціальних програм – *майстрів створення запитів*. Режим конструктора дозволяє повністю створювати будь-який запит, проте це потребує певних умінь від користувача. Майстер автоматично створює запит, який можна видозмінити, відкривши його в режимі конструктора.

Створення запиту розпочинається з вибору однієї або декількох таблиць, на основі яких буде створюватись запит. Залежно від завдання, яке повинен виконувати запит, користувач з полів таблиці створює структуру результуючої динамічної таблиці. Після цього створюються умови, які будуть виконувати фільтрування, сортування, заміну чи обчислення у записах таблиці.

Запит на вибірку

Запит на вибірку дозволяє вибрати з бази даних інформацію, яка цікавить користувача в даний момент. Іноді при створенні запитів необхідно використовувати поля, які належать відразу декільком таблицям бази даних. Такі запити називають реляційними. Їх можна використовувати для створення віртуальних таблиць, тобто набору даних, що належать різним таблицям, але виглядають як одна таблиця.

Програма *майстра* запускається після вибору ярлика *Создание запроса с помощью мастера* на вкладці *Запросы*.

На першому кроці майстра вибирають таблицю чи запит та поля, які будуть брати участь у пошуку даних. Для вибору таблиці (чи існуючого запиту), розкривають список *Таблицы и запросы*. Після цього у поле *Доступные поля* помістяться усі його поля. Для вибору полів, які братимуть участь у роботі створюваного запиту, потрібно кнопками із зображенням стрілок перенести потрібні поля в область *Выбранные поля*. Після вибору усіх потрібних полів для нового запиту натискають кнопку *Далее*.

У другому діалоговому вікні майстра створення запитів у полі *Задайте имя запроса* вводять ім'я створюваного запиту та вибирають одну із опцій продовження роботи після натискання кнопки *Готово: Открыть запрос для просмотра данных* (тобто виконати запит) чи *Изменить макет запроса* (тобто відкрити конструктор запитів для внесення змін у структуру запиту).

Для створення запитів з складною структурою використовують *конструктор*, який завантажується за допомогою ярлика *Создание запроса в режиме конструктора* на вкладці *Запросы* вікна бази даних. На екрані з'являється два вікна: вікно конструктора запитів *Запрос1: запрос на выборку* та вікно *Добавление таблицы*, яке містить три вкладки – *Таблицы*, *Запросы*, *Таблицы и запросы*.

У вікні *Добавление таблицы* на відповідних вкладках знаходиться перелік наявних об'єктів бази даних. Для вибору таблиць та запитів, на основі полів яких створюватиметься новий запит, потрібно по черзі виділяти об'єкти курсором миші та клацати на кнопці *Добавить*. Після кожного такого вибору у верхній частині вікна конструктора запитів з'являтиметься невелике вікно із назвою вибраного об'єкту. Завершити процес вибору об'єктів треба за допомогою кнопки *Заккрыть*.

У верхній частині вікна конструктора запиту розміщуються вікна з полями таблиць, на основі яких створюються запити, у нижній частині знаходиться бланк запиту, в якому задають критерії відбору даних, та параметри запиту, а також спосіб відображення полів на екрані.

Наступним етапом створення запиту є заповнення бланку конструктора запиту. У перший рядок бланку (*Поле*) розміщують назви полів, які братимуть участь у відборі та аналізі даних. При цьому у наступному рядку (*Имя таблицы*) автоматично розміститься назва таблиці, поле якої входить до структури запиту. Третій рядок бланку запиту (*Сортировка*) дозволяє задати параметр сортування даних поля результуючої таблиці запиту (доступні параметри знаходяться у списку, який відкривається після його активізації). Оскільки сортування можна здійснювати одночасно по декількох полях, то встановлення параметрів сортування можливе для кожного поля у бланку запиту. Для прискорення сортування поля, дані яких необхідно відсортувати, варто розмістити поруч.

Рядок бланку *Вывод на экран* призначений для встановлення опції, яка виводитиме, чи не виводитиме вибране поле в динамічну таблицю. За замовчуванням після включення поля таблиці у бланк запиту, у рядку *Вывод на экран* встановлюється значок 0, який означає, що поле виводитиметься у таблицю з вибраними записами. Для того, щоб його зняти, достатньо клацнути на ньому лівою кнопкою миші – він перетвориться у порожній квадратик.

Наступні рядки бланку запиту (*Условие отбора* та *или*) призначені для створення запиту, який відбиратиме лише такі записи таблиці, які задовольняють певні умови. Сюди вводять умови, на основі яких здійснюватиметься відбір даних.

Увівши всі потрібні параметри у бланк запиту, його закривають та у вікні *Сохранение* присвоюють ім'я.

Редагувати запит можна лише в режимі конструктора.

Для того, щоб *додати* у запит поле, перетягують його зі списку полів вікна таблиці, яке знаходиться у верхній частині конструктора запиту, у відповідну колонку бланку конструктора запиту. При цьому поле, яке там знаходилося, а також усі поля правіше від нього, переміщуються вправо.

Для *вилучення* поля зі структури запиту, виділяють його та натискають клавішу *Del* або виконують команду *Правка* → *Удалить*.

Для *переміщення* поля на інше місце використовують технологію перетягування об'єктів. Після переміщення інші поля також можуть зміститися.

Запити з параметрами

Запити, які відрізняються між собою лише різними значеннями параметрів, називаються *запитами з параметрами*. Переважно потреба у таких запитах виникає, коли треба виконувати послідовно запити однакової структури, які відрізняються лише параметрами у критеріях відбору.

Для створення параметричного запиту у рядку *Условие отбора* бланку конструктора вводять не конкретне значення параметра, а текст у квадратних дужках (наприклад, *[Значення параметра]*). Цей текст буде відображатися у діалоговому вікні *Введите значение параметра* після запуску запиту. Користувачу потрібно у поле введення цього діалогового вікна ввести конкретне значення параметра і після цього запит відобразить відібрані записи.

Перехресний запит

Якщо потрібно об'єднати дані у вигляді двомірної таблиці, створюють *перехресні запити*. Такі запити компактно відображають дані та об'єднують однорідну інформацію. Таблиці, в яких відображаються результати виконання перехресного запиту, зручні для створення діаграм та графіків.

Створити перехресний запит можна за допомогою майстра або в режимі конструктора. Для використання майстра потрібно у діалоговому вікні *Новый запрос (Вставка* → *Запрос)* вибрати елемент *Перекрестный запрос*.

На першому кроці майстра треба вибрати об'єкт (таблицю чи запит) на основі якого буде створюватиметься запит. На другому і третьому кроці – вибирають назви полів, які повинні стати заголовками стовпців та рядків відповідно. При

цьому на зразку відображатиметься макет майбутньої динамічної таблиці.

На наступному кроці майстра задається функція, з допомогою якої здійснюватиметься аналіз даних.

Для проведення у запиті підсумкових обчислень за вибраною функцією для кожного рядка включають опцію *Вычислить итоговое значение для каждой строки*. При цьому у перехресний запит майстер помістить стовпчик *Итоговое значение_Назва*, де *Назва* – назва поля, зі значеннями якого проводяться обчислення за вибраною функцією.

На останньому кроці майстра задають ім'я створюваного запиту та вибирають спосіб відображення його після завершення роботи з майстром (перегляд результатів у вигляді динамічної таблиці чи перегляд структури у режимі конструктора).

Запит на зміну даних

За допомогою *запитів на змінювання* користувач може змінювати дані в таблиці, додавати, знищувати групу записів, створювати нові таблиці. Відміна цих запитів від запитів на вибірку полягає в тому, що вони не просто фільтрують дані, а виконують певні дії над відфільтрованими даними. Запити на змінювання найчастіше використовуються для знищення, поновлення чи архівування записів, а також для виконання обчислень.

Використовуючи *запит на оновлення*, користувач може змінити групу записів, яка відібрана на основі певних критеріїв. Ці запити корисні у випадку внесення змін у певний набір даних.

До запитів на змінювання належать запити на *доповнення* однієї таблиці записами іншої. Вони можуть працювати як з таблицями однієї бази даних, так і декількох. Таблиці при цьому не обов'язково мають ідентичну структуру, але типи полів повинні бути однаковими.

Запити на знищення ліквідують в таблиці всі записи, які відповідають певній умові. Такі запити також часто використовуються для архівування даних.

Запит на створення нової таблиці дозволяє з динамічної таблиці створити нову таблицю бази даних. Часто такі запити використовуються для архівування або створення резервних копій таблиць.

Для створення *запиту на зміну даних* у таблиці зі списку кнопки *Тип запроса* вибираємо елемент *Обновление* або виконуємо команду меню *Запрос → Обновление*. Після цього з бланку запиту зникнуть рядки *Сортировка* і *Вывод на экран*, але з'явиться рядок *Обновление*. Бланк конструктора заповнюють лише тими полями таблиці, дані яких потрібно змінити. У рядку *Обновление* записують обчислювальний вираз, за яким будуть змінюватись дані.

Після виконання запиту на оновлення дані відповідного поля зміняться і відновити стару інформацію вже буде неможливо.

У випадку необхідності збереження результату виконання запиту у вигляді окремого об'єкту бази даних (таблиці), створюють *запит на збереження таблиці*. Для цього, створивши відповідний запит у режимі конструктора, у списку кнопки *Тип запроса* панелі інструментів програми вибирають елемент *Создание таблицы* або виконують команду меню *Запрос → Создание таблицы*. У діалоговому вікні, яке при цьому відкриється, вводять назву майбутньої таблиці та задають її місце розташування.

Іноді виникає необхідність об'єднання записів декількох таблиць в одну. Таку операцію може виконати *запит на об'єднання записів*. Для того, щоб додати записи однієї таблиці до іншої, потрібно на основі однієї з таблиць створити запит на вибірку, у який увійдуть всі поля таблиці, та задати відповідний тип запиту (командою *Запрос → Добавление* або вибравши елемент *Добавление* списку кнопки *Тип запроса*).

У діалоговому вікні, яке після цього відкриється, вибирають назву таблиці, до якої потрібно долучити записи, та встановлюють місце розташування вибраної таблиці. Після закриття діалогового вікна на бланку конструктора з'явиться новий рядок *Добавление* та назви полів таблиці, в які будуть переноситись дані з кожного поля.

Запит на знищення використовують, якщо потрібно з існуючої таблиці вилучити записи, які задовольняють певним критеріям. Для цього у вікні конструктора створюють запит на вибірку із заданими критеріями, а потім вибирають тип запиту за допомогою команди *Запрос → Удаление* або відповідного елемента списку кнопки *Тип запроса*. При цьому у бланку конструктора запиту з'являється рядок *Удаление* з однаковим значенням у всіх колонках бланку.

Після запуску на виконання такого запиту, програма видасть повідомлення про кількість записів, які будуть знищені, та попередить про те, що їх не можна буде відновити. Якщо користувач погоджується з таким вибором, то натиснення на кнопку *Да* інформаційного діалогового вікна завершить процес виконання запиту.

Форми

Якщо інформація у базі даних часто поновлюється, то доцільно використовувати форми. Форми – це своєрідні бланки-формуляри, які відображають інформацію таблиць та допомагають вносити зміни та доповнення в неї.

Один з різновидів форм, – *кнопкові форми* – можуть використовуватись для вибору інших форм, попереднього перегляду та друкування звітів, тобто для автоматизації роботи з базою даних.

Будь-яка форма будується на основі даних полів таблиці. Якщо ж форма повинна містити дані полів декількох зв'язаних таблиць, то можна спочатку створити запит, а вже потім, на основі полів запиту – форму.

На екрані кожну форму можна відобразити:

- у режимі форми;
- у режимі таблиці;
- у режимі конструктора.

Перехід між цими режимами відбувається за допомогою команд пункту меню *Вид* або за допомогою кнопки *Вид* панелі інструментів.

Для відображення, додавання, знищення чи оновлення даних використовують *режим форми* або *режим таблиці*, а для проектування та редагування структури об'єкту використовують *режим конструктора*.

У програмі передбачено створення форми одним із трьох способів:

- на основі шаблону;

- за допомогою майстра створення форми;
- у режимі конструктора форм.

Вибрати спосіб створення форми можна за допомогою ярликів вкладки *Формы* вікна бази даних або у діалоговому вікні *Новая форма*, яке відкривається кнопкою *Создать* панелі інструментів вікна бази даних.

У вікні *Новая форма* у полі *Выберите в качестве источника таблицу или запрос* рекомендується зразу ж вибрати зі списку таблицю або запит, на основі якої створюватиметься форма. Після цього вибирають один із можливих способів створення форми.

Програма Access дозволяє створювати форми на основі існуючих шаблонів – *автоформ*. Автоформи поділяються на три типи: у стовпчик, в рядок та таблична.

Для *автоформи в стовпчик* характерним є те, що для кожного запису таблиці відводиться окрема сторінка форми. Вона особливо добре підходить для записів з великою кількістю полів.

Автоформа в рядок розміщує кожен запис у окремий рядок. Цей вид форми дозволяє наочно подати декілька записів на сторінці.

Автоформа таблична за своїм виглядом аналогічна до звичайної таблиці.

Для створення автоформи потрібно у діалоговому вікні *Новая форма* вказати назву об'єкту, на основі якого створюватиметься форма та вибрати тип шаблону. Після натискання кнопки *ОК* на екрані з'явиться готова форма. У ній знаходитимуться усі поля об'єкту, проте послідовність розміщення полів може не співпадати із послідовністю в таблиці.

У програмі Access можна створювати різноманітні форми: *прості, складні, з діаграмами, у вигляді зведених таблиць*, а також *кнопкові та форми-заставки*.

У програмі Access передбачена можливість створення *складної форми*, яка відображає записи зв'язаних таблиць. Для кожного запису головної форми з'являється один або декілька записів підпорядкованої форми. Складні форми відображають дані таблиць, які зв'язані співвідношенням *один-до-багатьох*, причому головна форма відображає записи таблиці, яка знаходиться у цьому співвідношенні зі сторони *один*, а підпорядковані та додаткові – зі сторони *багато*.

Така форма може бути подана у вигляді однієї форми (головної, в якій вбудовані підпорядковані або у вигляді декількох зв'язаних форм, кожна наступна з яких відкривається за допомогою вбудованої у форму кнопки чи вимикача.

У програмі Access передбачена можливість створення ланцюжка зв'язаних підлеглих форм (до семи рівнів вкладеності), причому кожна із зв'язаних форм може мати як просту, так і досить складну структуру.

Перегляд записів у формі відбувається за допомогою рядка *Запись*, який розміщений у нижній частині вікна форми, а також за допомогою клавіш *PgUp* і *PgDn*. Перехід між полями здійснюється за допомогою клавіші *Tab* або клавіш керування курсором.

Для *введення* у форму лише *нових даних* доцільно скористатися командою меню *Записи → Ввод данных*, яка заховає усі існуючі записи, залишивши порожній бланк для введення нової інформації. Повернутися до режиму перегляду можна за допомогою команди *Записи → Удалить фильтр*.

Знищити запис з форми можна за допомогою клавіші *Del*, якщо попередньо позначити цей запис за допомогою маркера.

Характерні для форм маленькі поля значно ускладнюють введення та обробку довгих текстів.

Для введення та обробки довгого тексту можна використовувати вікно *Область ввода*, яке відкривається після того, як курсор встановлений у потрібне поле та натиснута комбінація клавіш *Shift+F2*.

У формах можна виконувати операції пошуку та заміни, а також використовувати фільтрацію для відбору записів, які задовольняють потрібні умови (аналогічно як і при роботі з таблицями).

Створення форми за допомогою майстра

Якщо потрібно створити форму за допомогою майстра, то у діалоговому вікні *Новая форма* необов'язково вибирати джерело інформації, оскільки майстер на першому кроці запропонує це зробити у полі *Таблицы и запросы*. На цьому ж кроці потрібно вибрати поля, дані з яких відображатимуться у формі.

При створенні форми, за допомогою якої дані таблиці будуть не лише переглядатись, а й доповнюватись чи поновлюватись, поля, які мають властивість *обязательное поле* повинні входити у форму.

Вигляд вікон наступних кроків майстра залежать від того, на основі однієї чи декількох таблиць створюватиметься форма.

При створенні форми на *основі полів однієї таблиці* другий та третій кроки майстра пропонують вибрати один із зовнішніх виглядів майбутньої форми (у стовпчик, в рядок, табличний чи вирівняний) та стиль оформлення. На останньому кроці, задають ім'я створеної форми та вибирають один із шляхів продовження роботи після завершення роботи майстра: відкрити форму для перегляду даних чи змінити макет форми за допомогою конструктора.

При створенні форми на *основі полів зв'язаних таблиць* слід, якщо форма повинна відображати значення полів декількох зв'язаних таблиць, то на другому кроці майстра створення форми задають спосіб відображення даних різних таблиць: у вигляді одинарної, підлеглих чи зв'язаних форм (цей вибір залежить від того, які функції повинна в майбутньому виконувати форма).

Для створення одинарної форми процес роботи майстра аналогічний до створення форми на основі однієї таблиці.

Для створення підлеглих форм у вікнах майстра задають вигляд підлеглої форми (в рядок чи табличний) та стиль оформлення головної форми. На останньому кроці задають назви головної та підлеглої форм та вибирають спосіб продовження роботи.

Для створення зв'язаних форм вибирають стиль оформлення обох форм та їх назви.

Після завершення роботи майстра на вкладці форм утвориться одна (у випадку одинарної) або дві (у випадку підлеглих та зв'язаних) форми.

Форма з діаграмою

Форма з діаграмою створюється для графічного представлення числових значень одного чи декількох полів таблиці. Перед початком побудови діаграми потрібно вибрати поля таблиці чи форми, значення яких будуть відображатися графічно. До уваги беруться лише дані, які задовольняють певним умовам. За замовчуванням значення стовпця утворюють на діаграмі ряд даних, а значення рядка – категорію. Діаграма може складатися з декількох рядів та категорій. Для відображення числових даних у програмі використовуються маркери – лінії, смуги, сектори тощо. На маркерах можуть знаходитися мітки, які відповідають числовим чи текстовим значенням. Для відображення діапазонів рядів даних, які представлено на діаграмі, використовуються осі. У легенді можна подати пояснення до позначень.

Найпростіший спосіб створення діаграми в Access – за допомогою майстра.

Для цього потрібно у діалоговому вікні *Новая форма* вибрати елемент *Диаграмма* і таблицю чи запит, на основі якої створюватиметься ця діаграма. Після цього запуситься майстер, що запропонує користувачеві вибрати ті поля, дані яких повинні бути відображені на діаграмі. Наступне вікно майстра дозволяє вибрати тип діаграми (у програмі передбачено 15 типів діаграм).

Після вибору типу діаграми майстер запропонує свій спосіб відображення даних на діаграмі. Користувач може змінити розташування полів у відповідних областях діаграми, перетягнути за допомогою миші кнопки з назвами полів, які розташовані у правій частині вікна, у відповідні області діаграми.

Необхідно зауважити, що при перетягуванні назви поля у область *Данные*, назва поля змінюється: перед ім'ям розміщується назва функції, за якою будуть проводитись обчислення. За замовчуванням – це функція сумування, тому поле називатиметься *Сумма_Назва поля*. Для зміни функції обчислення двічі клацають мишкою по назві обчислювального поля в області *Данные* і у діалоговому вікні з переліком функцій групових операцій, яке при цьому відкриється, вибирають потрібну функцію та натискають кнопку *ОК*. У полі *Данные* можуть знаходитись декілька обчислюваних полів, дані в яких розраховуються згідно різних функцій. Якщо область *Ряды* залишити незаповненою, тоді підписами у діаграмі будуть назви обчислювальних полів. У цьому ж вікні є кнопка *Образец*, яка дозволить у будь-який момент переглянути результати побудови діаграми.

У останньому вікні майстра задають назву діаграми, а також вибирають спосіб продовження роботи після завершення роботи майстра: відкрити форму з діаграмою або змінити структуру форми чи діаграми.

Звіти

Звіт – це кінцевий результат виконання задач системи управління базою даних. Звіти використовують для наочного представлення даних таблиці та для відображення зведеної інформації, яка отримана в результаті певних обчислень. Причому інформація може відображатися як у текстовому варіанті, так і за допомогою діаграм.

Звіти, які можна створити в Access, поділяються на категорії:

- *звіти у рядок* – дані відображаються по рядках, причому кожне поле бази даних займає окремий стовпець;
- *звіти у стовпець* – усі поля відображаються в один стовпець і вирівнюються по лівій межі; за своїм виглядом нагадують форми;
- *поштові наклейки* – звіти, які використовуються для друку поштових наклейок.

Програма Access передбачає створення звіту зі стандартною структурою, тобто у звіті зазвичай знаходиться основна частина та колонтитули. Проте користувач крім основної частини, може використовувати такі області як *заголовок* та *примітку звіту*, *верхній* та *нижній колонтитули* сторінки, а також *заголовок* та *примітку групи*. Застосування заголовку, примітки та колонтитулів сторінки аналогічне як у формах. У заголовку звіту зазвичай записують його назву, відомості про дату та час його створення. Примітка може містити підсумкове значення за цілим звітом. Ці області знаходяться лише на початку та в кінці звіту. У верхньому колонтитулі сторінки розміщують назви стовпців (полів), а у нижньому – номер сторінки. Ці колонтитули знаходяться на кожній сторінці звіту.

На екрані звіт можна відобразити лише у двох режимах: у режимі *перегляду* та в режимі *конструктора*. За замовчуванням після вибору піктограми звіту на вкладці *Отчеты*, звіт відображається у режимі перегляду, тобто у вигляді, який він матиме при друкуванні. Цьому ж режиму відповідає кнопка *Просмотр* панелі інструментів вікна бази даних.

Якщо потрібно відобразити структуру звіту у режимі конструктора, то можна скористатися кнопкою *Конструктор* панелі інструментів вікна бази даних.

Процес створення звіту подібний до створення форми.

В Access можна створювати звіти трьома способами: використовуючи функцію *автозвіт* (тобто за зразком існуючого шаблону), за допомогою *майстра* створення звітів або з використанням набору інструментів *конструктора* звіту.

Найшвидший спосіб створення – використання функції *автозвіт*, коли програма запропонує вибрати один із існуючих шаблонів: *в стовпчик* чи *в рядок*.

Майстер допомагає користувачеві створити структуру звіту, а саме вибрати поля з інформацією для аналізу, згрупувати та відсортувати дані, створити макет майбутнього звіту та задати параметри його друкування. Майстер дозволяє також задати параметри вигляду готового звіту, вибравши один із запропонованих стилів фону.

Зазвичай звіти створюються одним із двох згаданих вище способів. Проте, якщо створений звіт не задовольняє користувача, то він може його змінити вручну у вікні конструктора звітів, хоча це вимагає від користувача значних умінь та навиків.

Для створення звіту потрібно на вкладці *Отчеты* вікна бази даних треба за допомогою ярликів *Создание отчета в режиме конструктора* або *Создание отчета с помощью мастера* викликати вікно *Новый отчет*, де вибрати джерело, на основі якого будуватиметься звіт.

Для того, щоб створити звіт на базі існуючого шаблону, потрібно вибрати елемент *Автоотчет: в столбец* чи *Автоотчет: ленточный* у діалоговому вікні *Новый отчет*.

При натисканні кнопки *ОК* з'явиться вікно, в якому знаходитиметься створений за шаблоном звіт з назвою, яка ідентична назві джерела даних. Для того, щоб цей звіт опинився на вкладці звітів бази даних, потрібно при закритті вікна

звіту вибрати кнопку для його збереження.

За допомогою майстра можна створити такі звіти:

- звіт, який відображає дані полів запиту чи таблиці;
- звіт, у якому дані згруповані, та проведено підсумкові обчислення для кожної групи;
- звіт з діаграмою;
- поштові наклейки для розсилки кореспонденції.

Для створення звіту за допомогою майстра потрібно скористатися відповідним ярликом на вкладці *Отчеты* вікна бази даних або у діалоговому вікні *Новый отчет* вибрати елемент *Мастер отчетов*. У цьому ж вікні можна вибрати джерело даних, проте, як і при роботі з майстром форм, це можна буде зробити на першому кроці майстра. Тут потрібно вибрати усі поля, дані яких повинні, відобразитися у звіті.

Якщо у звіті братимуть участь поля зв'язаних таблиць, то програма автоматично групує дані відповідно до встановлених зв'язків. Якщо ж потрібно змінити структуру групування, то такі зміни можна внести на другому кроці майстра. Для цього додають або усувають якесь поле (за допомогою кнопок > та <) чи змінюють послідовність групування даних за допомогою кнопок *Уровень*.

Програма автоматично сортує дані у тих полях, де здійснюється групування. Якщо потрібно відсортувати дані й у інших полях, тоді у наступному діалоговому вікні майстра потрібно вказати назви цих полів та порядок сортування. Якщо у сортуванні беруть участь числові поля, то у цьому діалоговому вікні буде знаходитись кнопка *Итоги*, яка дозволяє відкрити інше діалогове вікно, в якому можна задати спосіб обчислення підсумкових значень для числових полів, де відобразяться всі числові поля, як беруть участь у звіті.

При обчисленнях можна використати такі групові операції як сумування, знаходження мінімального чи максимального значення, обчислення середнього арифметичного. У звіті можуть відобразитися дані з підсумками або лише підсумкові значення, а також передбачена можливість обчислення відсотків.

На наступному кроці майстра створення звіту користувач може вибрати спосіб подання даних у звіті, а саме:

- *ступенчатый* – дані розміщуються відповідно до рівнів групування, утворюючи «східці»;
- *блок* — дані розміщуються блоками;
- *структура 1* та *структура 2* – ступінчасте розміщення даних перекриттям;
- *по левому краю 1* та *по левому краю 2* – вирівнювання даних усіх рівнів по лівому краю.

У цьому ж вікні можна задати орієнтацію сторінки та задати опцію, яка задаватиме оптимальну ширину полів для того, щоб дані розмістилися на одній сторінці.

Перейшовши у наступне діалогове вікно, користувач зможе підібрати стиль оформлення звіту.

На останньому кроці потрібно присвоїти звіту ім'я та вибрати один із способів продовження роботи після завершення роботи майстра: перегляд звіту чи зміну макету звіту у режимі конструктора.

Для вибору чи формування стилю відображення даних у звіті потрібно, перебуваючи в режимі конструктора, виконати команду *Формат* → *Автоформат* і у діалоговому вікні, яке відкриється, вибрати потрібний стиль. У цьому ж вікні за допомогою кнопки *Настройка* можна змінити параметри вибраного стилю або створити новий на основі вибраного.

2. Приклад практичної роботи

Практичне завдання № 1 Створення бази даних та таблиць. Типи даних. Заповнення таблиць

Хід роботи:

1. Створити нову базу даних:

- запустити *Microsoft Access*;
- у вікні, що з'явиться вибрати “Новая база данных”
- у наступному вікні зазначити назву бази даних “Торгівельна фірма” і вказати папку, в якій буде зберігатися база даних;
- натиснути “Ок”;
- як результат з'явиться робоче вікно бази даних:

* при роботі з базами даних Access, об'єкти бази даних, такі як: таблиці, запити, звіти, форми зберігаються в одному файлі, який має розширення *.mdb;

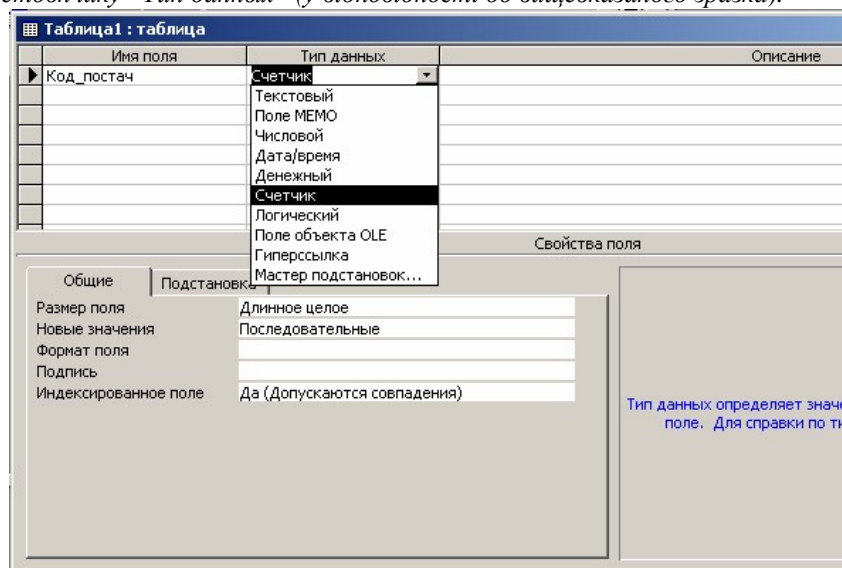
* створити базу даних можна іншим чином (з вікна Access) - виконати команду меню “Файл ⇒ Создать”. У вікні “Создание” вибрати вкладку “Общие” і натиснути на значку “База даних”. У вікні “Файл новой базы данных” задати ім'я файла – “Торгівельна фірма” та вказати папку для збереження цього файлу.

2. Створити таблицю “Постачальники” за наступним зразком

	Имя поля	Тип данных
🔍	Код поставч	Счетчик
	Назва	Текстовый
	Разр_рах	Числовой
	Адреса	Поле МЕМО
	Телефон	Текстовый

- на закладці “Таблицы” натиснути кнопку “Создать”;
- вибрати режим створення таблиці – “Конструктор”;

- натиснути “Ок”;
- у вікні конструктора таблиці вказати назви полів у стовпчику “Имя поля” та вибрати із списку типи даних для кожного з полів в стовпчику “Тип данных” (у відповідності до вищевказаного зразка):




- Для поля “Код_постач” встановити ознаку ключового поля:
 - встановити курсор на поле “Код_постач”;
 - на панелі інструментів натиснути кнопку “Ключевое поле” – ;
 - в результаті зліз від назви поля з’явиться зображення ключа.
- Для кожного із полів у відповідності з нижченаведеною таблицею на вкладці “Общие” вікна конструктора таблиці вказати необхідні параметри:

Таблица: **Постачальники**

Поле	Тип даних	Розмір (формат)	Кіл-ть десятк. знаків	Підпис
Код_постач	Счетчик	Длинное целое	-	Код поставщика
Назва	Текстовый	60	-	Назва поставщика
Розр_рах	Числовой	Длинное целое		Розрахунковий рахунок
Адреса	Поле МЕМО	-	-	Адреса
Телефон	Текстовый	20	-	Телефон

Поле	Индексированное	Обов’язкове	Умова на значення	Повідомлення про помилку
Код_постач	Да (Совпадения не допускаются)	-	-	-
Назва	Нет	Да	-	-
Розр_рах	Нет	Нет	-	-
Адреса	-	Нет	-	-
Телефон	Нет	Нет	-	-

- Зберегти створену таблицю:
 - вибрати пункт меню “Файл⇒Сохранить”;
 - у вікні “Сохранение” вказати ім’я таблиці – “Постачальники”

- натиснути "Ок";
 - закрити вікно конструктора таблиці.
6. Аналогічно створити інші таблиці по таблицях що наведені нижче:

Таблиця: **Покупці**

Поле	Тип даних	Розмір (формат)	Кіл-ть десятк. знаків	Підпис
Код_покуп	Счетчик	Длинное целое	-	Код покупця
Назва	Текстовый	60	-	Назва покупця
Адреса	Поле МЕМО	-	-	Адреса
Телефон	Текстовый	20	-	Телефон
Код_виду	Числовой	Длинное целое	-	Код виду покупця

Поле	Індексоване	Обов'язкове	Умова на значення	Повідомлення про помилку
Код_покуп	Да (Совпадения не допускаются)	-	-	-
Назва	Нет	Да	-	-
Адреса	-	Нет	-	-
Телефон	Нет	Нет	-	-
Код_виду	Да (Допускаются совпадения)	Да	>0	Код виду покупця повинен бути > 0!

Ключове поле – Код_покуп

Таблиця: **Види покупців**

Поле	Тип даних	Розмір (формат)	Кіл-ть десятк. знаків	Підпис
Код_виду	Счетчик	Длинное целое	-	Код виду покупця
Вид	Текстовый	25	-	Вид покупця

Поле	Індексоване	Обов'язкове	Умова на значення	Повідомлення про помилку
Код_виду	Да (Совпадения не допускаются)	-	-	-
Вид	Нет	Да	-	-

Ключове поле – Код_виду

Таблиця: **Агенти**

Поле	Тип даних	Розмір (формат)	Кіл-ть десятк. Знаків	Підпис
Код_агента	Счетчик	Длинное целое	-	Код поставщика
ПІБ	Текстовый	80	-	Прізвище Ім'я По-батькові
Дата_народ	Дата/время	Средний формат даты		Дата народження
Адреса	Поле MEMO	-	-	Домашня адреса
Тел_роб	Текстовый	20	-	Робочий телефон
Тел_дом	Текстовый	20	-	Домашній телефон
Стать	Логический	Да/Нет		Чоловіча/Жіноча
Фото	Поле объекта OLE	-	-	Фотографія

Поле	Індексоване	Обов'язкове	Умова на значення	Повідомлення про помилку
Код_агента	Да (Совпадения не допускаются)	-	-	-
ПІБ	Нет	Да	-	-
Дата_народ	Нет	Нет	-	-
Адреса	-	Нет	-	-
Тел_роб	Нет	Нет	-	-
Тел_дом	Нет	Нет	-	-
Стать	Нет	Нет	-	-
Фото	-	Нет	-	-

Ключове поле – Код_агента

Таблиця: **Товари**

Поле	Тип даних	Розмір (формат)	Кіл-ть десятк. знаків	Підпис
Код_тов	Счетчик	Длинное целое	-	Код товару
Назва_тов	Текстовый	100	-	Назва товару

Поле	Індексоване	Обов'язкове	Умова на значення	Повідомлення про помилку
Код_тов	Да (Совпадения не допускаются)	-	-	-
Назва_т	Нет	Да	-	-

ОВ				
----	--	--	--	--

Ключове поле – Код_тов

Таблиця: Угоди

Поле	Тип даних	Розмір (формат)	Кіл-ть десятк. знаків	Підпис
Код_уг	Счетчик	Длинное целое	-	Код угоди
Код_постач	Числовой	Длинное целое	-	Код поставщика
Код_покуп	Числовой	Длинное целое	-	Код покупця
Код_тов	Числовой	Длинное целое	-	Код товару
Код_агента	Числовой	Длинное целое	-	Код агента
Кіл_тов	Числовий	Длинное целое	0	Кількість товару
Дата	Дата/время	Краткий формат даты	-	Дата укладання угоди
Сума	Денежный	Денежный	2	Сума угоди
Код_вид_роз	Числовой	Длинное целое	-	Код виду розрахунку
Оплата	Логический	Да/Нет	-	Оплачено ТАК/НІ

Поле	Індексоване	Обов'язкове	Умова на значення	Повідомлення про помилку
Код_уг	Да (Совпадения не допускаются)	-	-	-
Код_постач	Нет	Да	>0	Код поставщика повинен бути > 0!
Код_покуп	Нет	Да	>0	Код покупця повинен бути > 0!
Код_тов	Нет	Да	>0	Код товару повинен бути > 0!
Код_агента	Нет	Да	>0	Код агента повинен бути > 0!
Кіл_тов	Нет	Да	>0	Потрібно занести кількість товару!
Дата	Нет	Да	-	-
Сума	Нет	Да	>0	Сума угоди не може бути

				від'ємно ю або нульово ю!
Код_вид_ро з	Нет	Да	>0	Код виду розрахун ку повинен бути > 0!
Оплата	Нет	Да	-	

Ключове поле – Код_уг

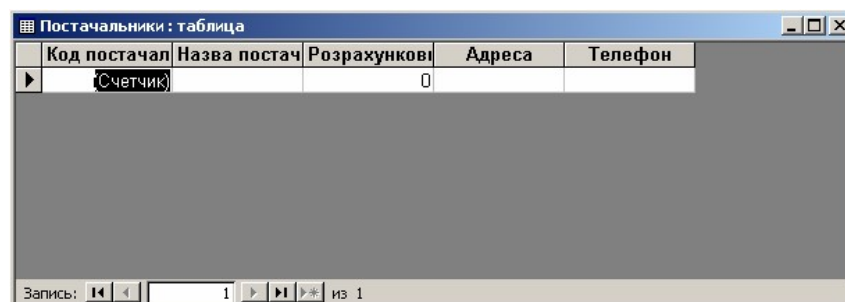
Таблиця: Види розрахунків

Поле	Тип даних	Розмір (формат)	Кіл-ть десятк. знаків	Підпис
Код_в_розрах	Счетчик	Длинное целое	-	Код виду розрахунку
Вид_роз	Текстовый	40	-	Вид розрахунку

Поле	Індексоване	Обов'язкове	Умова на значення	Повідомлення про помилку
Код_в_розрах	Да (Совпадения не допускаются)	-	-	-
Вид_роз	Нет	Да	-	-

7. Записати в таблицю “Постачальники” данні:

- на вкладниці “Таблицы” подвійним натисканням на імені таблиці (або вибрати необхідну таблицю та натиснути “Открыть”);



- у вікні даних таблиці занести наступні дані:

Таблиця: Постачальники

Код постачальника	Назва постачальника	Розрахунковий рахунок
1	АРТ-візаж	1254680213
2	Глобал-тренд	1136625014
3	Компас	1999850021

Код постачальника	Назва постачальника	Розрахунковий рахунок
4	Юніор	1456002350
5	Тандем	1223542209

Адреса	Телефон
м.Житомир, вул.Цюлковського 52	37-54-13
м.Київ, вул.Саксаганського 85, 14	252-43-95
м.Київ, вул.Шота-Роставелі 24, 13	264-33-87
м.Харків, вул.Червоноармійська 47, 71	64-27-13
м.Вінниця, вул.Чапаєва 29, 8	42-17-54

- після вводу даних закрити вікно таблиці.

8. По вищевказаному прикладі заповнити всі таблиці наступними даними:

Таблиця: **Покупці**

Код покупця	Назва покупця
1	Метеор
2	ТОВ Продукт
3	ЗАТ Товстун
4	ПП Торба
5	Окомелко О.
6	Грищенко В.
7	ПП Міщук
8	Пекар
9	ПП Булочка
10	Комелков В.

Адреса	Телефон	Код виду покупця
м.Житомир, вул.Щорса, 29	25-17-14	1
м.Житомир, вул.Якіра 12	20-42-86	1
м.Житомир, просп.Миру, 143	25-43-03	1
м.Житомир, пров.Лівий, 13	33-10-85	2
с.Великі Доставалки, вул.Надоїдлива, 2	2-22-22	3
смт.Черняхів, вул.Чапаєва, 26	2-24-31	3
с.Бородянка, вул.Красна, 47		2
м.Житомир, вул.Ватутіна, 257	20-85-61	1
м.Житомир, вул.Хлібна, 27	30-10-20	2
м.Житомир, пров.Тяжкий, 64	24-17-96	3

Таблиця: **Види покупців**

Код виду покупця	Вид покупця
1	Підприємство
2	Приватний підприємець
3	Фізична особа

Таблиця: **Агенти**

Код агента	Прізвище Ім'я По-батькові	Дата народження	Домашня адреса
1	Сидорчук Василь Петрович	03-Вер-71	м. Житомир, вул.Щорса 142
2	Носенко Степан Віталійович	20-Лют-77	м. Житомир, вул.Лятошинського

Код агента	Прізвище Ім'я По-батькові	Дата народження	Домашня адреса
			14а, 27
3	Легенька Валентина Олексіївна	25-Кві-80	м. Житомир, вул.Київська 117, 20
4	Хробуст Світлана Петрівна	04-Лис-76	м. Житомир, вул.М.Жукова 19, 48
5	Титаренко Володимир Михайлович	14-Тра-72	м. Житомир, вул.Малікова 83, 10
6	Хмеляр Зінаїда Василівна	27-Чер-74	м. Житомир, вул.Космонавтів 141, 15

Робочий телефон	Домашній телефон	Чоловіча / Жіноча	Фотографія
22-14-56	25-68-01	Да	
22-14-57	20-87-41	Да	
22-14-51	22-66-93	Нет	
22-14-49	26-57-30	Нет	
22-14-62	26-88-69	Да	
22-14-60	34-24-18	Нет	

Вставка даних в останній стовпчик – Фотографія (тип даних – Поле об'єкту OLE) проводиться наступним чином:

- встановити курсор на поле “Фотографія”;
- вибрати “Вставка ⇒ Об'єкт...”;
- у наступному вікні зазначити “Создать из файла” та за допомогою кнопки “Обзор” вибрати необхідну фотографію
- Натиснути “Ок”, в результаті у вибраному полі з'явиться напис “Точечный рисунок”.

Таблиця: Товари

Код товару	Назва товару
1	Борошно
2	Гречка
3	Сіль
4	Цукор
5	Печиво
6	Вафлі
7	Цукерки
8	Рис
9	Пшоно
10	Кава

Таблиця: Види розрахунків

Код виду розрахунку	Вид розрахунку
---------------------	----------------

Код виду розрахунку	Вид розрахунку
1	Безготівковий
2	Готівкою
3	Бартер

Таблиця: Угоди

Код угоди	Код постачальника	Код покупця	Код товару	Код агента	Кількість товару
1	5	10	10	6	100
2	2	4	6	4	25
3	3	5	7	3	20
4	4	8	6	3	30
5	5	3	5	1	150

Дата укладання угоди	Сума угоди	Код виду розрахунку	Оплаче но ТАК/Н І
10.03. 2002	2 000,00 грн.	1	Да
02.03. 2002	800,00 грн.	3	Нет
13.02. 2002	420,00 грн.	2	Да
12.03. 2002	780,00 грн.	1	Нет
27.02. 2002	2 100,00 грн.	1	Да

9. Самостійно доповнити таблицю “Угоди” до 50 записів.

- * Код постачальника повинен бути в межах від 1 до 5;
- * Код покупця повинен бути в межах від 1 до 10;
- * Код товару повинен бути в межах від 1 до 10;
- * Код агента повинен бути в межах від 1 до 6;
- * Код виду розрахунку повинен бути в межах від 1 до 3;
- * Дата укладання угоди – в межах I-го кварталу 2002 року;
- * Кількість товару, суму угоди та ознаку оплати встановити довільною.


Практичне завдання № 2

Зв’язування таблиць. Схема даних

Хід роботи:

10. Відкрити робочий файл бази даних “Торгівельна фірма”.

11. Перейти у вікно схеми даних для зв’язування таблиць між собою:

- виконати пункт меню “Сервис ⇒ Схема данных...” або натиснути на панелі інструментів кнопку ;
- у вікні “Добавление таблицы” вибрати таблицю “Покупці” та натиснути кнопку “Добавить”;
- вибрати іншу таблицю – “Види покупців” та натиснути “Добавить”.

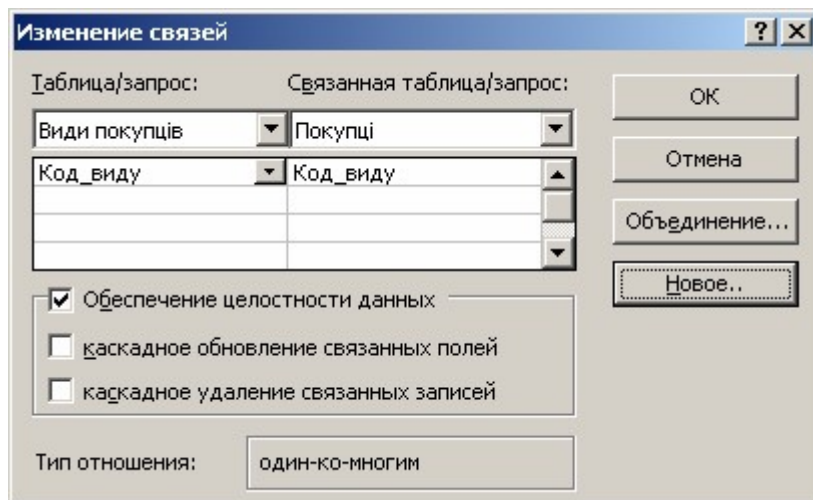
* після додавання таблиці у вікні схеми даних буде з’являтися зображення доданої таблиці

- натиснути кнопку “Закрыть”.

12. Встановити зв’язок між двома таблицями – “Покупці” і “Види покупців” по полях “Код_виду” та “Код_виду” відповідно:

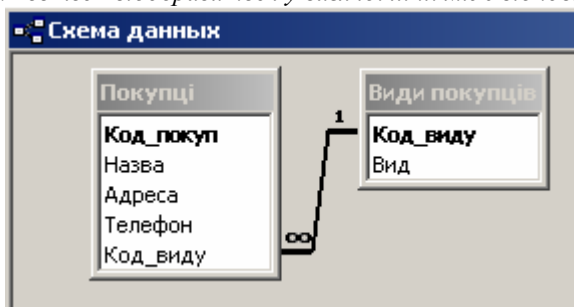
- виділити лівою кнопкою миші поле, по якому встановлюється зв’язок в першій таблиці (поле “Код_виду” таблиці “Покупці”);
- утримуючи кнопку миші натиснутою, перемістити курсор на потрібне поле другої таблиці (поле “Код_виду” таблиці “Види покупців”);

- у вікні, що з'явилося – “Изменение связей”, впевнитися в правильності встановлення зв'язку (імена полів та імена таблиць):



* ознака “Обеспечение целостности данных” повинна бути встановлена, що забезпечує коректність зв'язків.

- візуально у вікні “Схема даних” зв'язок відобразиться у вигляді лінії між відповідними полями зв'язаних таблиць.



13. Зберегти створену схему даних:

- виконати “Файл⇒Сохранить”.

14. Закрити вікно схеми даних.

15. Додати до схеми даних таблицю “Постачальники”:

- відкрити вікно схеми даних (див. п.2);
- на панелі інструментів натиснути кнопку “Отобразить таблицу” або натиснути правою кнопкою миші на вільному місці вікно схеми даних та вибрати в меню, що з'явиться пункт “Добавить таблицу”;
- у вікні “Добавление таблицы” вибрати потрібну таблицю – “Постачальники”;
- натиснути “Добавить”.

16. Аналогічно додати до схеми даних всі інші таблиці бази даних.

* у вікні “Добавление таблицы” можна виділяти декілька таблиць одночасно і, відповідно, одночасно додати декілька таблиць;

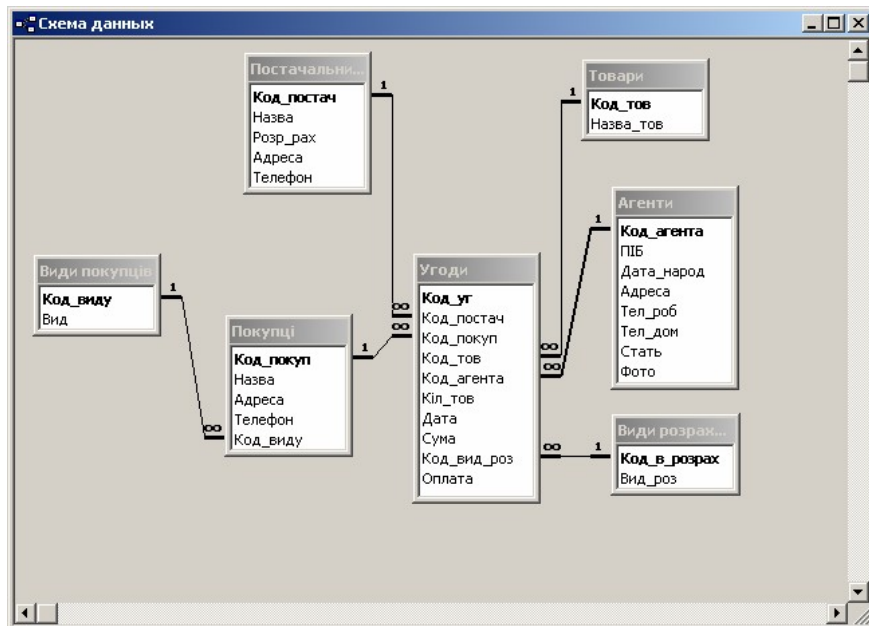
* потрібно слідкувати, щоб не додати одну і ту ж таблицю декілька разів;

* щоб вилучити з схеми даних помилково занесену таблицю, потрібно цю таблицю виділити (лівою кнопкою миші) та натиснути на клавіатурі Delete або виконати “Правка⇒Удалить”.

17. Користуючись вже відомим методом з'єднання таблиць, створіть зв'язки між всіма таблицями бази даних за наступними правилами:

- таблиця “Угоди” поле “Код_вид_роз” з таблицею “Види розрахунків” по полю “Код_в_розрах”;
- таблиця “Угоди” поле “Код_покуп” з таблицею “Покупці” по полю “Код_покуп”;
- таблиця “Угоди” поле “Код_постач” з таблицею “Постачальники” по полю “Код_постач”;
- таблиця “Угоди” поле “Код_тов” з таблицею “Товари” по полю “Код_тов”;
- таблиця “Угоди” поле “Код_агента” з таблицею “Агенти” по полю “Код_агента”.


18. Зберегти створену схему даних (яка повинна мати приблизно такий вид, який зображено на рис. нижче) та закрити вікно схеми даних



Практичне завдання № 3 Побудова запитів



Хід роботи:


19. Відкрити робочий файл бази даних "Торгівельна фірма".
 20. У вікні бази даних перейти на закладку "Запроси":
 21. Створити запит, який би виводив всіх поставальників (назви) та їх телефони:
- * перед створенням запиту спочатку потрібно проаналізувати завдання, тобто потрібно в'яснити на основі яких даних (таблиць) буде побудовано запит.
- у вікні бази даних на закладці "Запросы" натиснути кнопку "Создать";
 - у вікні "Новый запрос" вибрати "Конструктор" та натиснути кнопку "Ок";
 - в результаті з'явиться вікно конструктора запитів та вікно "Добавление таблицы", в якому потрібно вибрати таблицю (або декілька) з якої буде робитися вибірка даних (в нашому випадку – це таблиця "Поставальники");
 - натиснути кнопку "Добавить", після чого в верхній частині вікна конструктора запитів з'явиться зображення вибраної таблиці:
- * якщо запит буде використовувати декілька таблиць, то всіх їх потрібно вибрати у вікні "Добавление таблицы" та додати до запиту.
- натиснути кнопку "Закрыть";
- * в результуючий набір даних (нижня частина вікна конструктора запиту) потрібно по черзі вивести ті поля, які містять необхідні дані (у нашому випадку – поле "Назва" та поле "Телефон"),
- двічі натиснути на потрібному полі ("Назва") у зображенні таблиці (в верхній частині вікна конструктора), **або** перенести лівою кнопкою миші необхідне поле у перший вільний стовпчик нижньої частини вікна конструктора, **або** в рядку "Поле" у списку, що з'явиться після натискання мишою, вибрати необхідне поле;
 - аналогічно у вихідний набір даних додати поле "Телефон";
 - в результаті вікно конструктора буде мати наступний вид:

- після створення запиту його потрібно протестувати, тобто натиснути кнопку “Запустити”  на панелі інструментів;
- на екрані з’явиться результат виконання запиту:



Запрос1 : запрос на выборку

Назва постач	Телефон
▶ АРТ-візаж	37-54-13
Глобал-тренд	252-43-95
Компас	264-33-87
Юніор	64-27-13
Тандем	42-17-54
*	

Запись:  1  из 5

* якщо результат виконання запиту неправильний – потрібно змінити структуру запиту, тобто повернутися до вікна конструктора запиту при допомозі натискання кнопки “Вид”  після чого відбудеться перехід до конструктору.

- після перевірки на правильність дії запиту необхідно його зберегти, для чого вибрати пункт меню “Файл⇒Сохранить”;
- у вікні “Сохранение” ввести назву запиту – “Телефони поставщиков”:

Сохранение  

Имя запроса:

Телефоны поставщиков

ОК

Отмена

- натиснути “Ок”;
- закрити вікно запиту.

22. Аналогічно створити наступні запити:

- **“Перелік покупців”** (містить дані: назва покупця, телефон покупця та його адреса; відсортувати по адресі покупців по зростанню);

* ознаку сортування встановлюють в рядку “Сортировка” нижньої частини вікна конструктора запиту:

Запрос1 : запрос на выборку

Покупці

*
Код_покуп
Назва
Адреса
Телефон

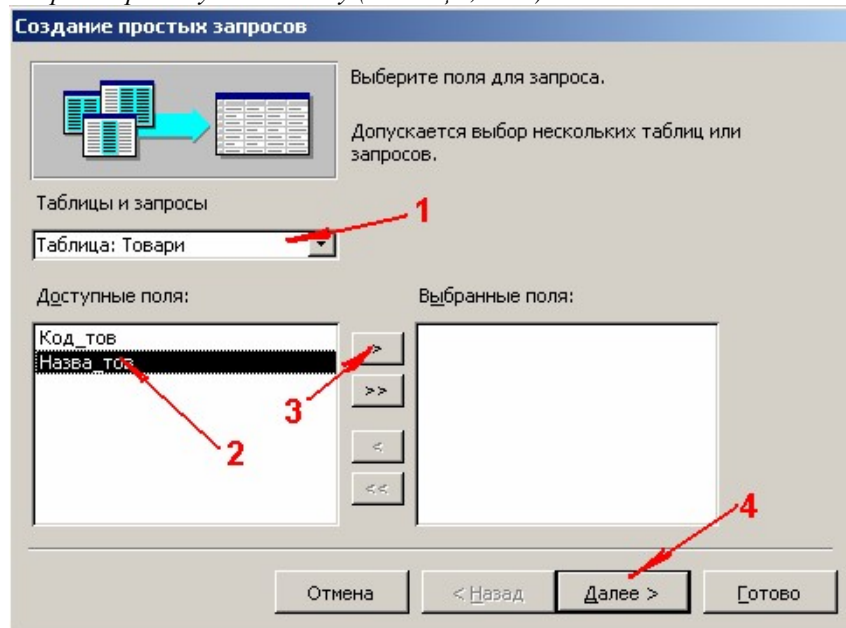
Поле:	Назва	Телефон	Адреса
Имя таблицы:	Покупці	Покупці	Покупці
Сортировка:			по возрастанию
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	по убыванию
Условие отбора:			(отсутствует)
или:			

- **“Роб_телефоны агентів”** (містить дані: ПІБ агента, робочий телефон агента; відсортувати по ПІБ агентів по зростанню);
- **“Перелік агентів”** (містить дані: ПІБ агента, дата народження, домашній телефон, адреса, стать; відсортувати по ПІБ агентів по зростанню);

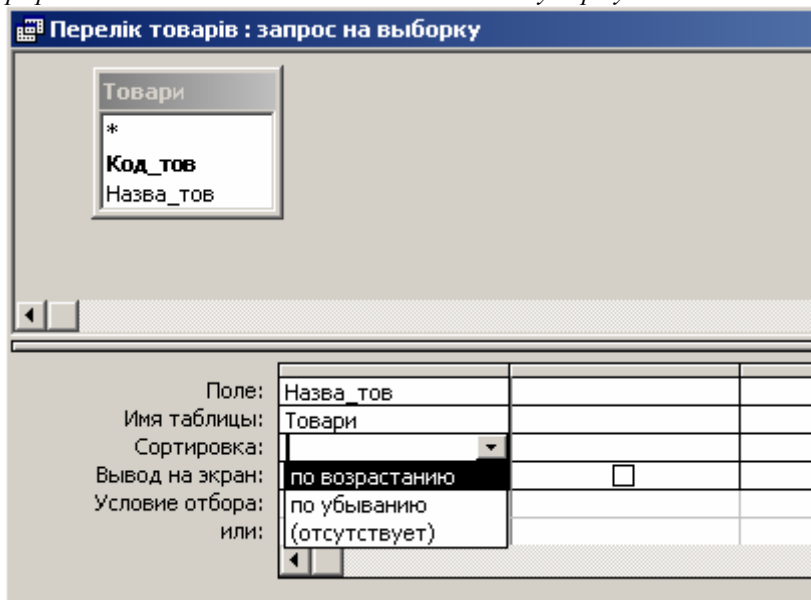
23. Створити запит **“Перелік товарів”** (містить дані: найменування товару) при допомозі методу “Простой запрос”

- На закладці запиту вибрати “Создать”;
- У вікні “Новый запрос” вибрати “Простой запрос”:

- вибрати необхідні параметри побудови запиту (таблицю, поля):



- вказати ім'я запиту – “Перелік товарів”;
 - натиснути “Готово”.
24. Аналогічно створити запит “Перелік видів розрахунків”, який містить вид розрахунків.
25. Для запиту “Перелік товарів” встановити ознаку сортування по назві товару по зростанню:
- вибрати необхідний запит для модифікації (“Перелік товарів”);
 - натиснути кнопку “Конструктор”;
 - у вікні конструктор зробити необхідні зміни – встановити ознаку сортування:



- закрити вікно конструктора, зберігши зміни.
26. Для запиту “Перелік видів розрахунків” встановити ознаку сортування по виду розрахунку по зростанню.
27. Створити запит “Покупці по виду”, який містить дані про вид покупця (юридична, фізична особа), назву покупця та його адресу:
- при допомозі конструктора створити новий запит, додавши до нього потрібні таблиці – “Покупці” та “Вид покупців”
- * перед створенням запиту потрібно проаналізувати де містяться необхідні дані – в яких таблицях та в яких полях, щоб в подальшому будувати запит на вже відомих таблицях!
- * при створенні запиту на основі декількох таблиць, зв'язки між ними будуть показуватися автоматично, якщо ці зв'язки були раніше побудовані в схемі дані.
- вибрати необхідні поля – “Вид” з таблиці “Вид покупців”, “Назва” та “Адреса” з таблиці “Покупці”;
 - встановити сортування по виду покупців по зростанню та вторинну ознаку сортування – по назві покупця по зростанню;
 - зберегти запит під ім'ям “Покупці по виду” та закрити вікно конструктору.
28. Аналогічно на основі декількох таблиць створити наступні запити:

- запит “Суми по постачальниках” містить дані: назва постачальника, сума угоди та дата угоди, сортування встановити по сумі угоди по зменшенню;
- запит “Оплата по покупцях” містить дані: назва покупця, дата угоди, ознака оплати сортування встановити по назві покупця по зростанню;
- запит “Угоди по агентах” містить дані: ПІБ агента, сума угоди, ознака оплати, сортування встановити по сумі угоди по зменшенню та по ПІБ агента по зростанню;
- запит “Суми по видах розрахунку” містить дані: сума угоди, ознака оплати, вид розрахунку, сортування встановити по сумі угоди по зменшенню.

3. Завдання до лабораторної роботи

1. Повторити теоретичний матеріал з курсу інформатики по СУБД MS Access.
2. Виконати та проаналізувати практичну роботу, що пропонується учням.
3. В ході аналізу сформулювати всі знання та вміння, що формуються у учнів даною практичною роботою.
4. Розробити фрагменти конспектів уроку пояснення нового матеріалу з теми, враховуючи знання і вміння, якими повинні оволодіти учні при вивченні СУБД MS Access.
5. Розробити систему контрольних питань та завдань для перевірки матеріалу теми.
6. Оформити звіт по лабораторній роботі.

Навчальна алгоритмічна мова.

ОСНОВНА МЕТА	Методика вивчення навчальної алгоритмічної мови в школі.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ	Знати основні методичні та фактологічні особливості вивчення навчальної алгоритмічної мови. Вміти розв'язувати практичні задачі на НАМ.
ППЗ	Виконавець "Черепиха".

1. Методичні рекомендації

На початку слід розглянути з учням основні поняття – "алгоритм", "алгоритмічна мова", "виконавець", "програма" і т.д., використавши шкільний посібник і наведені нижче означення.

Алгоритм – скінчений набір правил, які дозволяють чисто механічно розв'язати будь-яку конкретну задачу із деякого класу однотипних задач.

Алгоритмічна мова – набір символів і система правил складання і тлумачення мовних конструкцій із цих символів, призначених для запису алгоритмів.

Програма (грецьке – оголошення, розпорядження) – опис алгоритму розв'язування задачі, викладений мовою, однозначно тлумачною виконавцем.

Програмування – процес складання програми.

Виконавець – об'єкт (автоматичний пристрій, ЕОМ і т.п.) чи суб'єкт, що виконує команди, вказані в програмі.

Кожний виконавець може виконувати команди тільки з деякого строго заданого списку - системи команд виконавця. Для кожної команди повинні бути задані умови придатності (у яких станах середовища може бути виконана команда) і описані результати виконання команди.

Людина аналізує поставлену задачу і складає послідовність команд, що ведуть до цілі. Ця послідовність утворює алгоритм, який людина передає ЕОМ. Зворотний зв'язок з ЕОМ на цьому етапі відсутній.

ЕОМ, виконуючи алгоритм, управляє виконавцем. При цьому виникає зворотній зв'язок ЕОМ із виконавцем. Найважливіша особливість цього етапу – відсутність зв'язку з людиною.

На ЕОМ цілком поширюється принцип формального виконання: ЕОМ не аналізує зміст алгоритму, вона послідовно, команда за командою, виконує його. Звідси випливає необхідність точної, що не припускає двозначності, форми запису алгоритму. Таку форму забезпечують формальні мови, одна із яких - навчальна алгоритмічна мова.

З принципу формального виконання випливає, що ні виконавець, ні ЕОМ не можуть чинити помилок. Всі помилки, пов'язані з керуванням виконавцями, здійснює людина.

Для опису будь-якого алгоритму використовуються **базові структури**: *лінійна, розгалуження та циклічна*.

Прикладом лінійної структури можуть бути правила запису арифметичних виразів в алгоритмічній мові і використання арифметичних виразів у якості аргументів.

Оскільки ЕОМ виконує дії формально, необхідний формальний засіб запису виразів. У алгоритмічній мові в якості такого засобу прийнятий лінійний запис. Крім формалізації запису й обчислення виразів, лінійний запис забезпечує можливість введення арифметичних виразів із клавіатури.

У арифметичні вирази можуть входити виклики функцій.

При обчисленні значень виразів ЕОМ діє по формальних правилах:

1. Замінити усі вхідні у вирази величини їхніми значеннями.
2. Обчислити значення виразів, що містяться у дужках. (Порядок дій всередині дужок визначається пунктами 2-6.)
3. Обчислити значення функцій.
4. Виконати справа ліворуч піднесення до степеня.
5. Виконати зліва праворуч множення і ділення.
6. Виконати зліва праворуч додавання і віднімання.

Команди розгалуження разом із командами повторення складають повний набір засобів конструювання алгоритмів. У цілому команди розгалуження простіше команд повторення і самі по собі засвоюються легше. Більш складно навчитися використовувати їх у сполученні з командами повторення.

Коли цикл використовується усередині однієї із серій команди якщо, складність виникаючої алгоритмічної структури може бути зменшена за рахунок застосування допоміжних алгоритмів.

Основна маса алгоритмів, що використовуються на практиці, має можливість "підлаштуватися" під конкретні умови при їх виконанні, тобто не тільки, наприклад, "підставляти" конкретні значення змінних в жорсткий ланцюг обчислень, але і змінювати хід самих обчислень в залежності від певних умов, вказаних через значення аргументів або проміжних результатів обчислень.

Прикладом такого «підлаштування» може бути алгоритм обчислення коренів квадратного рівняння

$$ax^2 + by + c = 0 \quad (1)$$

Як відомо із шкільного курсу математики, корені рівняння x_1, x_2 існують, якщо дискримінант $d > 0$ або $d = 0$, і не існують якщо $d < 0$:

$$d = b^2 - 4ac ;$$
$$x_1 = \frac{-b + \sqrt{d}}{2a} ; x_2 = \frac{-b - \sqrt{d}}{2a} \quad (2)$$

Таким чином, при розв'язуванні квадратного рівняння слід спочатку обчислити дискримінант d за формулою (1), а потім, якщо $d > 0$ або $d = 0$, приступати до обчислення коренів x_1, x_2 за формулою (2). У випадку, якщо $d < 0$, слід повідомити про те що розв'язків немає, і припинити подальші дії.

Цей приклад слід детально розглянути на занятті.

Перед початком самостійної роботи слід нагадати учням відповідний теоретичний матеріал, вид опису команди розгалуження навчальною алгоритмічною мовою, повну і скорочену форми команди розгалуження.

Слід підкреслити, що результатом перевірки логічних умов є бульова величина, яка набуває лише двох можливих фіксованих значень – «істина» (true) і «хиба» (false). Ці значення на практиці часто замінюються відповідно значеннями «так» і «ні» або «1» і «0».

Традиційна форма конструкції «розгалуження» має вигляд :

Якщо < умова > То < оператор > Інакше < оператор >

Найпростіша з команд повторення – команда **Цикл-N раз**.

При вивченні цього матеріалу слід розкрити сутність команди повторення Цикл-N раз як складеної команди. Показати на прикладах форму запису команди і специфіку її застосування в алгоритмах. Учні повинні знати призначення команди повторення Цикл-N раз, її загальний вид; уміти записувати, використовувати і виконувати алгоритми з цією командою.

Складені команди (повторення і розгалуження) грають в алгоритмічній мові особливу роль. Вони не задають конкретних дій виконавця, як, наприклад, лінійні команди. Виконуючи ці команди, ЕОМ визначає послідовність виклику простих команд, що входять до них. У такий спосіб забезпечується можливість багатократного безпомилкового повторення серій команд, вибору потрібної команди в залежності від умов. Саме ці можливості лежать в основі масових застосувань ЕОМ.

Освоєння команди Цикл-N раз повинно допомогти учням зрозуміти специфіку складених команд і загальні риси команд повторення.

Потрібно звернути увагу на важливу відмінність команди повторення від лінійних команд. У алгоритмі, складеному з лінійних команд, по кожній команді можна точно й однозначно зазначити, яка команда повинна здійснюватися слідом за нею. Щоб визначити черговий виклик в алгоритмі з командою повторення, потрібно знати, чи закінчилося її виконання.

Команда повторення часто сприймається учнями лише як засіб скорочення запису алгоритму, що містить повторювані дії. Проте основна необхідність у ній виникає в тому випадку, коли треба зазначити число повторень у загальному виді.

В шкільному посібнику вивчається команда повторення *Цикл-Поки*. При вивченні цієї команди розглядається важлива ідея зворотного зв'язку, передача інформації від виконавця до ЕОМ. Використання зворотного зв'язку дозволяє складати гнучкі алгоритми керування виконавцями. У команді *Цикл-Поки* застосовуються операція перевірки умов.

При вивченні цього матеріалу слід ввести команду повторення *Цикл-Поки*; на прикладах показати її застосування. Ввести поняття зворотного зв'язку, простих і складених умов, показати способи застосування умов у команді *Цикл-Поки*.

Учні повинні знати загальний вид команди повторення *Цикл-Поки*, її властивості і схему виконання. Учні повинні вміти записувати команду повторення *Цикл-Поки*, розуміти зміст складених умов і способи їх запису; уміти записувати і використовувати складені умови, що містять 1-2 прості умови.

При виклику команди зворотного зв'язку виконавець виконує відповідну дію і передає ЕОМ деяку інформацію. Нагадаємо, що нас цікавить результат, а не механізм виконання команди. Результат виконання команди зворотного зв'язку - інформація, отримана ЕОМ. Ця інформація подана у вигляді величини певного типу.

Сформулюємо деякі властивості команди *Цикл-Поки*:

- 1) умова перевіряється тільки перед виконанням тіла циклу, але не перевіряється в процесі виконання;
- 2) якщо умова не дотримується із самого початку, тіло циклу не виконується жодного разу;
- 3) після завершення команди *Цикл-Поки*, незалежно від числа повторень, умова не дотримується;
- 4) виконання команди *Цикл-Поки* може не завершитися.

Зупинимось на основній відмінності команди *Цикл-Поки* від інших команд повторення.

Число повторень у циклах *Цикл-N раз* і *Цикл-Для* визначається по заголовку і, отже, точно відомо момент початку виконання циклу. Число повторень у циклі *Цикл-Поки* по заголовку не визначається і стає відомим тільки після виконання циклу.

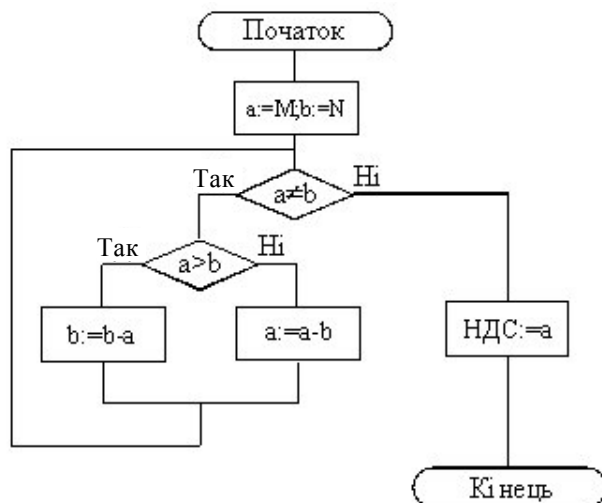
При машинному варіанті навчання доцільно використати виконавець "Черепашка". При цьому на початку уроку слід пригадати вивчені раніше теоретичні положення: для чого потрібна організація циклу в програмах; коли потрібно використовувати конструкції **Цикл-N раз** та **Цикл-Поки**. Слід роз'яснити, що аналогом циклу **Цикл-N раз** є команда "повтор", а команди **Цикл-Поки** – команда "цикл".

Учні повинні розуміти, що конструкція "цикл" є універсальною, яка дозволяє описувати довільні цикли як з наперед відомою, так і невідомою кількістю повторень.

Прикладом використання циклу з відомою кількістю повторень може бути алгоритм обчислення факторіала деякого числа n , тобто добутку чисел від 1 до n ($1!=1$, $2!=2$, $3!=6$, $4!=24$, $5!=120$ і т.д.), описаний мовою "Черепашка" (програма T6.TUR).

Інший приклад – алгоритм знаходження найбільшого спільного дільника (НСД) двох натуральних чисел M і N (програма T9.TUR), поданий блок–схемою на рисунку.

Задано M і N



Тут наперед невідома кількість повторень, оскільки вона залежить від конкретних значень аргументів і співвідношень між ними, тому для організації циклу використовується конструкція "цикл".

Зауважимо, однак, що і в першому прикладі алгоритм обчислення факторіала може бути реалізований з використанням конструкції "цикл":

```

старт
aa:=1
bb:=1
цикл bb<15
[
aa:=aa*bb
bb:=bb+1
йти 32,9*bb+10
виведення aa
]
стоп
    
```

Таким чином, конструкція "повтор" використовується тільки тоді, коли на момент початку циклу відома кількість повторень. Конструкція "цикл" є універсальною, але, як правило, використовується в тих випадках, коли число повторень (наприклад, ітерація при розв'язуванні трансцендентних рівнянь методом послідовних наближень) невідоме заздалегідь, і умова виходу із циклу обчислюється безпосередньо в тілі циклу.

Необхідно врахувати, що якщо умова виходу із циклу настає до початку його виконання, то тіло циклу не виконується жодного разу.

Для прикладу обчислимо суму всіх натуральних чисел від $bb=1$ до $ee=15$.

```

старт
aa:=0
bb:=1
dd:=1
ee:=15
cc:=""
kk:="СУМА="
цикл bb<=ee
[
aa:=aa+bb
bb:=bb+dd
йти 32,30
виведення aa
виведення cc
]
виведення kk
виведення aa
виведення cc
стоп
Попробуйте змінити рядки 3 і 5
bb:=7
ee:=5
    
```

При виконанні алгоритмів бажано користуватися покроковим режимом для унаочнення процесу послідовного виконання алгоритму: перевірки умови, виконання тіла циклу, повернення до початку циклу, виходу із циклу.

Далі слід виконати з учнями такі завдання.

ЗАВДАННЯ

1. Скласти алгоритм розв'язування кубічного рівняння

$x^3 + 2x^2 - x - 2 = 0$ методом ітерацій на відрізок $[0;2]$ з точністю 0.005.

2. Виконати алгоритм в покроковому режимі. Скласти алгоритм наближеного обчислення означеного інтеграла від функції

$f(x) = x^2 - 2x + 2$ методом прямокутників, поділивши інтервал $[0;2]$ на 50 частин.

Для першого завдання корисно виконати 2–3 обчислення при різних значеннях точності і прослідкувати, як при цьому змінюється час розв'язування задачі. Якщо дозволяє час, можна задати іншу функцію, що є монотонною і змінює свій знак на інтервалі $[a; b]$).

2. Завдання до лабораторної роботи

1. Опрацювати відповідний матеріал за шкільними посібниками [5] та [8].
2. Виконати завдання 1 і 2, що пропонуються учням.
3. Скласти конспект уроку по базових структурах навчальної алгоритмічної мови. В конспекті обов'язково висвітлити:
 - загальний запис команд базових структур;
 - блок-схеми команд базових структур;
 - демонстраційні приклади.
4. Дати тлумачення алгоритму та сформулювати вимоги до алгоритмів.
5. Оформити звіт по лабораторній роботі.

Мова програмування BASIC
Лінійне програмування числових виразів

ОСНОВНА МЕТА	Методика вивчення операторів і функцій лінійного програмування. Формування навичок складання алгоритмів на мові BASIC.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ	Знати основні принципи, команди, оператори та функції лінійного програмування числових виразів на мові BASIC. Вміти розв'язувати практичні задачі лінійного програмування.
ППЗ	Інтерпретатор BASIC.

1. Методичні рекомендації

Перетворення даних у Бейсику провадяться за допомогою операцій і виразів. Відповідно до типу результуючого значення і типом операндів операції виразу розрізняються по типу - числові (чи арифметичні), рядкові і логічні. Перетворення значень виконуються з урахуванням наступних правил:

- значення, що присвоюється змінній, перетворюється до типу цієї змінної;
- у ході обчислень при виконанні кожної операції над операндами, представленими в ЕОМ з різною точністю, значення операнда з меншою точністю перетворюється до значення з більшою точністю. Більшу точність представлення має і результат операції;
- при виконанні логічних операцій операнди завжди перетворюються до цілочисленному представлення. Результат операції також є цілим числом;
- перетворення значення з більшою точністю представлення до значення з меншою точністю виконується шляхом округлення;
- не допускається присвоєння числового значення рядковій змінній і рядкового значення числовій змінній.

Вираз являє собою складену за визначеними правилами комбінацію констант, змінних, елементів масивів і звертань до функцій, з'єднаних знаками операцій.

Під константою розуміється задане в явному вигляді значення, що використовується в програмі. Константи розрізняються за типом. У стандартному Бейсику існують два типи констант: числові і рядкові. У Турбо-Бейсику і Квік-Бейсику є ще один тип даних – іменовані константи, що подібно змінним мають ім'я, але не можуть змінювати значення в ході виконання програми.

Числові константи являють собою додатні і від'ємні дійсні числа. При записі додатних чисел поставлення знака «плюс» необов'язкове.

Змінна – це іменована величина, що може приймати різні значення в процесі виконання програми. Змінні розрізняються по типу в залежності від типу величин, значення яких вони можуть приймати. У стандартному Бейсику використовуються величини чотирьох типів: цілочисленні, дійсні одинарної точності, дійсні подвійної точності і рядкові; у Турбо-Бейсику поряд із зазначеними типами використовуються довгі цілі змінні.

Тип змінної може бути заданий за допомогою використання спеціальних знаків наприкінці імені змінної, що є показниками типів:

- \$ – для рядкової,
- % – для цілочисленної,
- ! – для дійсної змінної одинарної точності,
- # – для дійсної змінної подвійної точності,
- & – для довгої цілочисленної змінної.

Якщо тип змінної не заданий явно, то за замовчуванням змінній присвоюється тип дійсної змінної одинарної точності. Помітимо, що знак, що вказує на тип змінної, є невід'ємним елементом імені, і тому імена A%, A\$, A!, A# і A& відповідають різним змінним.

Типу відповідає формат збереження її значення в пам'яті ЕОМ. Значення рядкової змінної зберігається в (4+n) байтах пам'яті (n – кількість символів у рядку).

- Типи змінних можуть бути задані оператором оголошення типу змінних:
- DEFINT– оголошення цілочисленних змінних;
 - DEFLNG– оголошення довгих цілочисленних змінних;
 - DEFSNG– оголошення дійсних змінних одинарної точності;
 - DEFDBL– оголошення дійсних змінних подвійної точності;
 - DEFSTR– оголошення рядкових змінних.

У Бейсику можна задавати і використовувати одномірні і багатомірні масиви. Імена масивів створюються по тим же правилам, що й імена змінних. Звертання до елемента масиву здійснюється шляхом вказівки імені масиву і значень індексів. Значення індексів записуються в круглих дужках після імені масиву і розділяються між собою комами. Допускаються завдання значення індексу у вигляді арифметичного виразу.

Оголошення масиву певного типу, вказівка імені, кількості індексів і їхніх максимальних значень провадиться оператором DIM.

Найбільш вживані операції реалізовані в Бейсику у вигляді функцій, що є складовою частиною мови. Функції можуть використовуватися в будь-якій місці, де можуть використовуватися змінні, за винятком лівої частини оператора присвоєння. Звертання до функції здійснюється шляхом вживання у виразах імені функції з вказівкою аргументів і параметрів, від яких вона залежить. При обчисленні виразів, у які входить звертання до функції, провадиться підстановка на місце виклику функції значення, що повертається нею. Значення, що повертається, може бути, як числовим, так і рядковим. В останньому випадку ім'я функції закінчується знаком \$. Не допускається звертання до рядкових функцій у числових виразах, і навпаки.

У числових виразах використовуються арифметичні операції, операції порівняння, логічні операції і виклики числових функцій.

Дії, що містять вирази, виконуються в наступному порядку:

- визначаються значення функцій;
- виконуються арифметичні операції відповідно до їхніх пріоритетів;
- виконуються операції порівняння;
- виконуються логічні операції відповідно до їхніх пріоритетів.

Операції з одним значенням пріоритету виконуються в порядку їхнього запису у виразі зліва направо. Для зміни порядку виконання операцій використовуються круглі дужки. Операції в дужках виконуються першими.

У таблиці перераховані всі арифметичні операції в порядку їхніх пріоритетів.

Знак	Назва операції	Приклади
\wedge	Піднесення до степеня	A^C ; a^2 ; $8^{2.11}$
$-$	Зміна знака	$-A$; -4
$*$, $/$	Множення, ділення	$A*B$; f/h ; $5.1*9/3.7$
\backslash	Цілочислен не ділення	$A\backslash B$; $7\backslash 8$
MOD	Ділення по модулю	$A \text{ MOD } B$; $77 \text{ MOD } 12$
$+$, $-$	Додавання, віднімання	$A+B$; $JJ-17.3$

Операція цілочисленного ділення виконується в такий спосіб: спочатку знаменник і чисельник округляються до цілочисленних значень, потім виконується звичайна операція ділення і, нарешті, дробова частина отриманого відкидається.

Операція ділення по модулю в такий спосіб: спочатку виконується цілочисленне ділення, потім береться залишок, що одержаний при діленні.

Програма на Бейсику являє собою послідовність операторів, що вказують, у якому порядку і які дії над даними чи апаратними засобами ЕОМ повинні бути виконані. Запис оператора звичайно містить у собі одне (рідше декілька слів) ключове слово, набір виразів, що визначають параметри оператора і набір операндів, тобто констант і змінних, над якими виконуються перетворення.

Звичайно на одному рядку розташовується один оператор, але і бувають випадки, коли оператор розташовується на декількох рядках. На один рядок можна також записувати декілька операторів, розділених між собою двокрапкою. Оператор може бути позначений міткою, що дозволяє робити на нього посилання з інших місць програми. У стандартному Бейсику як мітки використовуються номери рядків. У Турбо-Бейсику і Квік-Бейсику крім натуральних чисел як мітки можна використовувати довільні рядки, що складаються по тим же правилам, що й імена змінних. Така мітка розташовується на окремому рядку перед рядком, що містить оператор, на який може бути зроблена посилання, і слідом за нею ставиться двокрапка.

2. Завдання до лабораторної роботи

1. Самостійно опрацювати:
 - запис констант, змінних, стандартних функцій;
 - правила запису арифметичних виразів;
 - арифметичний оператор присвоєння;
 - організацію найпростішого введення-виведення даних.
2. Розробити програму обчислення виразу відповідно до варіанту (див. Додаток).

Програма повинна розпочинатися відповідними реквізитами. Наприклад,

Варіант 3. Студент Петренко В.Г.

Значення змінних для обчислення повинні вводитися з клавіатури за допомогою оператора INPUT з певним запрошенням. Наприклад,

Введіть значення a, b, c:

Виведення значень обчислень повинно супроводжуватися відповідним поясненням. Наприклад,

Для $a = 7.8$; $b = -0.56$; $c = 456$ значення обчислень $z = 1.23456$; $Q = -4.56789$

3. Розробити фрагмент конспекту уроку пояснення нового матеріалу щодо початків програмування на мові BASIC:
 - структура і побудова програми;
 - запуск на виконання;
 - налагодження програми;
 - опис найважливіших операторів і функцій з прикладами їх використання (\sqrt{x} , $\sqrt[n]{x}$, e^x , $\sin x$, $\cos x$, $\operatorname{tg} x$, $\ln x$, $|x|$, PRINT, INPUT);
 - демонстраційна програма лінійного програмування.
4. Розв'язати задачі.

Задача 1.

```
10 DEFINT A-C
20 A#=2.718281828
30 REM C=A#
40 A=15+C
50 B%=5: A#=A#+1
60 A%=A%+B
```

Яких значень набудуть змінні
A#, C, B, A в результаті
виконання програми?

Задача 2.

```
10 A#=6.2222222222
20 B#=1.3333333333
30 B=B#: REM A!=A#
40 A%=A#+B#
50 B=B%+B
60 B%=B+B%
```

Яких значень набудуть змінні
A%, A!, B%, B в результаті
виконання програми?

5. Дати відповідь на контрольні питання.
6. Оформити звіт по лабораторній роботі.

Контрольні питання

1. Вказати старшинство виконання операцій при обчисленні арифметичних виразів.
2. Як організувати виведення значень змінних у супроводі найменування цих змінних?
3. Дати визначення лінійному обчислювальному процесу.
4. Які типи числових констант використовуються в мові Бейсик і як вони записуються?
5. Які є стандартні скалярні типи?
6. Як у програмі задаються типи змінних?
7. Які стандартні функції використовуються в мові Бейсик?
8. Якими операторами здійснюється введення даних і як виконуються ці оператори?
9. Для чого використовуються і як записуються коментарі у програмі?

Додаток

Варіант	Вираз	Значення змінних
1.	$a = \frac{2 \cos(x - \frac{\pi}{6})}{\frac{1}{2} + \sin^2 y};$ $b = 1 + \frac{z^2}{3 + \frac{z^2}{5}}$	$x = 1,426;$ $y = -1,220;$ $z = 3,5$
2.	$a = \left x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right ;$ $b = (y - x) \frac{y - \frac{z}{y-x}}{1 + (y - x)^2}$	$x = 1,825;$ $y = 18,225;$ $z = -3,298$

3.	$y = e^{-bt} \sin(at + b) - \sqrt{ bt + a };$ $S = b \sin(at^2 \cos 2t) - 1$	$a = -0,5;$ $b = 1,7;$ $t = 0,44$
4.	$W = \sqrt{x^2 + b} - \frac{b^2 \sin^3(x + a)}{x};$ $y = \cos^2 x^3 - \frac{x}{\sqrt{a^2 + b^2}}$	$a = 1,5;$ $b = 15,5;$ $x = -2,9$
5.	$S = \frac{x^3}{tg^2(x + b)^2} + \frac{a}{\sqrt{x + b}};$ $Q = \frac{bx^2 - a}{e^{ax} - 1}$	$a = 16,5;$ $b = 3,4;$ $x = 0,61$
6.	$R = \frac{x^2(x + 1)}{b} - \sin^2(x + a);$ $S = \sqrt{\frac{xb}{a}} + \cos^2(x + b)^3$	$a = 0,7;$ $b = 0,05;$ $x = 0,5$
7.	$y = \sin^3(x^3 + a)^2 - \sqrt{\frac{x}{b}};$ $z = \frac{x^2}{a} + \cos(x + b)^3$	$a = 1,1;$ $b = 0,004;$ $x = 0,2$
8.	$y = \sqrt[3]{mtgt + c \sin t };$ $z = m \cos(bt \sin t) + c$	$m = 2;$ $c = -1;$ $t = 1,2;$ $b = 0,7$
9.	$y = btg^2 x - \frac{a}{\sin^2 \frac{x}{a}};$ $d = ae^{-\sqrt{a}} \cos \frac{bx}{a}$	$a = 3,2;$ $b = 17,5;$ $x = -4,8$
10.	$f = \ln(a + x^2) + \sin^2(x / b);$ $z = e^{-cx} \frac{x + \sqrt{x + a}}{x - \sqrt{ x - b }}$	$a = 10,2;$ $b = 9,2;$ $x = 2,2;$ $c = 0,5$

Мова програмування BASIC
Лінійне програмування графічних об'єктів

ОСНОВНА МЕТА	Ознайомлення з операторами і функціями лінійного програмування графічних об'єктів. Формування навичок складання алгоритмів на мові BASIC.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ	Знати основні принципи, команди, оператори та функції лінійного програмування графічних об'єктів на мові BASIC. Вміти розв'язувати практичні задачі.
ППЗ	Інтерпретатор BASIC.

1. Методичні рекомендації

Графічні засоби Бейсика дозволяють створювати складні зображення, формовані з безлічі точок. Точки (пікселі) розташовуються на екрані у вигляді прямокутної матриці, утворюючи горизонтальні рядки і вертикальні стовпці. Відлік стовпців (координата x) ведеться зліва праворуч, а рядків (координата y) – зверху вниз. Крайня ліва верхня точка має координати (0,0). Світна точка екрана може характеризуватися кольором і яскравістю. Кількість точок залежить від роздільної здатності екрана і графічного адаптера.

У Бейсику можна налаштовуватися на різні режими роботи дисплея, параметри яких визначаються застосовуваним графічним адаптером (CGA, VGA, EGA, MDA). Установка режиму роботи дисплея провадиться оператором SCREEN:

SCREEN [<режим>] [, <колір>] [, [<активна_сторінка>] [, <відображувана_сторінка>]]

<Режим> – числовий вираз, значення якого задає режим роботи екрана. Припустимі значення: 0, 1, 2, 3, 7, 8, 9, 10, 11, 12, 13. Значення 0 задає текстовий режим роботи екрана, інші – графічний. У стандартному Бейсику параметр < режим > може приймати лише значення 0, 1, 2.

<Колір> – числовий вираз зі значенням 0 чи 1. Цей параметр включає чи відключає кольорове зображення. В усіх графічних режимах, крім режиму 1, це опція ігнорується. У текстовому режимі (0) значення опції < колір >, рівна 0, означає використання чорно-білого зображення, і всі інші значення – кольорового. У режимі 1, навпаки, значення опції <колір>, рівне 0, означає використання кольорового зображення;

<Активна_сторінка> – числовий вираз, значення якого вказує номер сторінки буфера екрана, куди будуть розміщуватися дані при виконанні операторів виводу на екран;

<Відображувана_сторінка> – числовий вираз, значення якого вказує номер сторінки буфера екрана, що буде відображатися на екран. Відображувана й активна сторінки можуть не збігатися. Ці параметри можуть приймати цілочисленні значення від 0 до 7. Максимальне припустиме значення номера сторінки визначається розміром екранної пам'яті і може бути рівним 0, 1, 3 чи 7. За замовчуванням номера активної і відображуваної сторінок встановлюються рівними нулю.

Формат застосування інших графічних операторів див. в допомозі до програми QBASIC.

2. Завдання до лабораторної роботи

1. Намалювати *змістовний* рисунок засобами BASIC.

Програма повинна використовувати такі оператори:

SCREEN (режим 12, екран 640 x 480 точок, 16 кольорів),
LINE (прямі і прямокутники),
CIRCLE (кола і еліпси),
DRAW,
PAINT (в тому числі зафарбовування шаблоном),
PSET,
LOCATE,
COLOR.

Рисунок повинен бути підписаний певним (не білим) кольором.

7. Розробити фрагмент конспекту уроку пояснення нового матеріалу щодо лінійного програмування графічних об'єктів на мові BASIC:
 - структура і побудова програми;
 - запуск на виконання;
 - налагодження програми;
 - опис найважливіших операторів і функцій (див. вище) з *прикладом* їх використання.
3. Дати відповідь на контрольні питання.
4. Оформити звіт по лабораторній роботі.

Контрольні питання

1. Дати характеристику графічним екранам.
2. Які процедури призначені для ініціалізації графічного режиму і які параметри вони мають?

3. Пояснити систему координат графічного екрана.
4. Якими операторами зображуються лінії на екрані дисплея?
5. Дати характеристику лініям, що зображуються на екрані дисплея?
6. Якими операторами відображаються прямокутники на екрані дисплея?
7. Якими операторами можна побудувати багатокутники?
8. Які оператори використовуються для побудови дуг, еліпсів і кіл?
9. Якими операторами можна встановлювати кольори зображень, екран дисплея?
10. Якими функціями здійснюється встановлення стилю заповнення замкнутих областей?

Мова програмування BASIC
Розгалужене та циклічне програмування

ОСНОВНА МЕТА	Ознайомлення з операторами і функціями розгалуженого та циклічного програмування. Формування навичок складання алгоритмів на мові BASIC.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ	Знати основні принципи, команди, оператори та функції розгалуженого та циклічного програмування на мові BASIC. Вміти розв'язувати практичні задачі розгалуженого та циклічного програмування.
ППЗ	Інтерпретатор BASIC.

1. Методичні рекомендації

Схема, що визначає порядок виконання операторів програми, називається структурою програми.

При структурі, що розгалужується, є один чи трохи операторів, що виконуються (чи не виконуються) у залежності від істинності заданої умови. Циклічна структура забезпечує повторне виконання групи операторів задане число раз.

Зміна природного порядку виконання операторів у Бейсику може бути організоване за допомогою чотирьох груп операторів:

- 1) оператори безумовної передачі керування (безумовного переходу);
- 2) умовні оператори;
- 3) оператори передачі керування по обчисленню;
- 4) оператори передачі керування в зв'язку з настанням заданого стану апаратних засобів.

В усіх версіях Бейсика є умовний оператор наступного формату:
IF <умова> THEN <оператор> [:<оператор>...][ELSE <оператор> [:<оператор>...]]

<Умова> - довільний логічний вираз; <оператор> – будь-який оператор Бейсика.

Результат виконання оператора залежить від значення логічного виразу. Якщо умова істинна, то виконуються оператори, записані після ключового слова THEN, після чого відбувається перехід до виконання наступного оператора. У випадку хибності умови виконуються оператори, записані після ключового слова ELSE, якщо цей фрагмент присутній в умовному операторі. Нарешті, якщо фрагмент ELSE відсутній, і умова помилкова, то відразу ж виконується наступний оператор.

Організація циклічного виконання операторів, повторюваних до досягнення змінної циклу заданого значення, провадиться у всіх версіях Бейсика оператором циклу FOR/NEXT.

FOR <змінна циклу> =X TO Y [STEP Z]

<тіло циклу>

NEXT <змінна циклу>

де <змінна циклу> – деяка змінна цілочисленного чи типу дійсного одинарної точності, X, Y, Z – арифметичні вирази, значення яких є початкове значення змінної циклу, кінцеве значення і збільшення змінної циклу (крок) відповідно, <тіло циклу> – рядки програми з операторами, що виконуються в циклі.

Виконання оператора циклу починається з присвоєння змінної циклу зазначеного початкового значення. Потім виконуються оператори, що складають тіло циклу. У результаті виконання оператора NEXT до поточного значення змінної циклу додається збільшення, після чого отримане значення порівнюється з кінцевим значенням. Якщо значення змінної циклу виявиться більше кінцевого значення, то цикл закінчується і керування передається оператору, що стоїть за оператором NEXT. У протилежному випадку здійснюється чергове виконання операторів, що складають тіло циклу. Так виконується оператор циклу у випадку додатного кроку. Якщо ж крок від'ємний (що припустимо), то в ході виконання циклу значення змінної циклу зменшується і вихід з циклу відбувається, коли змінна циклу стане менше кінцевого значення. Фрагмент STEP може бути опущений і тоді за замовчуванням крок циклу передбачається рівним 1.

Якщо параметри циклу задані не константами, а більш складними арифметичними вираженнями, то перед початком циклу визначаються їхні значення і надалі вони не коректуються, навіть якщо в тілі циклу міняються значення змінних, що входять у ці вирази. Тіло циклу не виконується жодного разу, якщо між значеннями параметрів циклу мають місце наступні співвідношення:

$$X > Y, Z > 0 \text{ чи } X < Y, Z < 0.$$

Для економії часу виконання програми змінну циклу і параметри варто задавати за допомогою цілочисленних змінних.

У тілі одного циклу може знаходитися інший цикл, що називається вкладеним. При наявності структури з вкладеними циклами всі змінні циклів повинні бути різні. Якщо вкладені цикли закінчуються в одному, то послідовність операторів NEXT може бути замінена на один оператор NEXT, у якому перераховані через кому всі змінні циклів.

Ім'я змінної циклу в операторі NEXT може бути опущено, при цьому оператор NEXT вважається відповідним найближчому оператору FOR. Для вкладених циклів указівка змінних циклів в операторі NEXT обов'язкова.

Передача керування ззовні усередину циклу, минаючи оператор FOR неприпустимий. У Турбо-Бейсику і Квік-Бейсику можливо дострокове закінчення виконання циклу по оператору EXIT FOR.

Оператор WHILE/WEND у всіх версіях мови призначений для організації циклу з передумовою, у якому оператори, що складають тіло циклу, виконуються циклічно доти, поки не стане рівним нулю заданий як умова виходу арифметичний вираз. Формат оператора такий:

WHILE <арифметичний вираз>

<тіло циклу>

WEND

У Турбо-Бейсику можна організувати достроковий вихід з циклу за допомогою оператора EXIT WHILE.

Допускається використання вкладених циклів WHILE/WEND. Якщо в програмі буде виявлений оператор WHILE без відповідного йому (найближчого у вкладених циклах) оператора WEND, чи, навпаки – WEND без WHILE, буде видане повідомлення про помилку.

2. Завдання до лабораторної роботи

1. Розробити програму обчислення виразів (2 завдання) відповідно до варіанту (див. Додаток лаб. роб. №3), використовуючи оператори розгалуження та циклу FOR ... NEXT, WHILE ... WEND, IF ... THEN ... ELSE....

Програма повинна розпочинатися відповідними реквізитами. Наприклад,

Варіант 3. Студент Петренко В.Г.

Значення змінних для обчислення повинні вводитись з клавіатури за допомогою оператора INPUT з певним запрошенням. Наприклад,

Введіть значення a, b, c:

Виведення значень обчислень повинно супроводжуватися відповідним поясненням. Наприклад,

Для **a = 7.8; b = -0.56; c = 456 найбільше число c = 456**

2. Розробити фрагмент конспекту уроку пояснення нового матеріалу щодо розгалуженого та циклічного програмування на мові BASIC:
 - структура і побудова програми;
 - запуск на виконання;
 - налагодження програми;
 - опис найважливіших операторів і функцій (див. вище) з *прикладми* їх використання.
3. Розробити завдання до контрольної роботи з даної теми, що включають 4 варіанти щодо розгалуженого та 4 варіанти щодо циклічного програмування на мові BASIC. Завдання представити з розв'язками в комп'ютерному вигляді.
4. Дати відповідь на контрольні питання.
5. Оформити звіт по лабораторній роботі.

Контрольні питання

1. Яка програма називається циклічною?
2. Як зображуються на блок-схемі алгоритм циклічної структури з відомим числом повторень?
3. Як реалізуються циклічні структури з відомим числом повторень на мові Бейсик?
4. Як зображуються на блок-схемі алгоритм циклічної структури з невідомим числом повторень?
5. Як реалізуються циклічні структури з невідомим числом повторень на мові Бейсик?
6. Що називають тілом циклу?
7. Що таке складений оператор і які функції він виконує?
8. Які обмеження накладаються на використання оператора FOR ... NEXT?
9. В чому суттєва відмінність між операторами циклу FOR ... NEXT та WHILE ... WEND ?
10. Яка відмінність у виконанні операторів IF ... THEN ... ELSE та IF ... THEN ?
11. Як за допомогою блок-схеми алгоритму відобразити структури розгалуження?
12. Якими командами мови Бейсик реалізується структура розгалуження?

ЛАБОРАТОРНА РОБОТА №19 (4 год.)

Мова програмування BASIC Обробка масивів та рядкових змінних

ОСНОВНА МЕТА	Ознайомлення з операторами і функціями обробки масивів та рядкових змінних. Формування навичок складання алгоритмів на мові BASIC.
ВИМОГИ ДО ЗНАНЬ І ВМІНЬ	Знати основні принципи, команди, оператори та функції обробки масивів та рядкових змінних на мові BASIC. Вміти розв'язувати практичні задачі обробки масивів та рядкових змінних.
ППЗ	Інтерпретатор BASIC.

1. Методичні рекомендації

Вираз називається рядковим, якщо результат перетворення даних, що вказується цим виразом, є рядком. У Бейсику є тільки одна операція над рядковими даними – операція конкатенації (зчеплення рядків), що позначається знаком плюс. У ході виконання цієї операції з двох рядків складається один новий рядок.

Для обробки рядкових даних у всіх версіях Бейсика є наступні оператори і функції: ASC, CHR\$, INSTR, LEFT\$, LEN, MID\$, RIGHT\$, SPACE\$ і STRING\$. У Турбо-Бейсику є ще функції LCASE\$ і UCASE\$, а в Квік-Бейсику – LCASE\$, UCASE\$, LTRIM\$ і RTRIM\$.

У мові Бейсик змінні рядкового типу описуються за допомогою команди DEFSTR або неявно шляхом використання суфікса \$ в іменах змінних. Наприклад,

DEFSTR A, B

або без попереднього опису

INPUT A\$

B\$="Бейсик".

Надати значення текстовим змінним (ввести дані) можна трьома способами за допомогою таких команд: 1) присвоєння; 2) INPUT; 3) READ.

Текстові дані можна виводити на екран дисплея, на друкуючий пристрій (або у файл даних) командою PRINT.

Набір текстових функцій, які входять до Бейсик-систем, наведемо в таблиці:

LEFT\$(A\$,K)	Відокремлює в A\$ частину рядка ліворуч довжиною K символів
RIGHT\$(A\$,K)	Відокремлює в A\$ частину рядка праворуч довжиною K символів
MID\$(A\$,L,M)	Відокремлює в A\$ частину рядка довжиною M символів, починаючи від символу з номером L
LEN(A\$)	Обчислює довжину текстового даного A\$
VAL(A\$)	Текстове дане перетворює у числове
STR\$(X)	Числове дане перетворює у текстове
INSTR(A\$,B\$)	Дає номер позиції, з якої починається входження тексту B\$ у текст A\$
CHR\$(<номер>)	Дає символ з таблиці ASCII з даним номером
ASC(<символ>)	Дає номер символу в таблиці кодів ASCII

Приклад. Нехай текстова змінна A\$ має значення "ІНФОРМАТИКА". Тоді зазначені функції дадуть такі результати:

Функції	Результат
LEFT\$(A\$,6)	ІНФОРМ
RIGHT\$(A\$,9)	ФОРМАТИКА
MID\$(A\$,6,3)	МАТ
LEN(A\$)	11
VAL("123")	123
STR\$(123)	"123"
INSTR(A\$,"Ф")	3
CHR\$(132)	"Д"
ASC("Д")	132

У Бейсику можна задавати і використовувати одномірні і багатомірні масиви. Імена масивів утворюються за тими ж правилами, що й імена змінних. Звертання до елемента масиву здійснюється шляхом вказівки імені масиву і значень

індексів. Значення індексів записуються в круглих дужках після імені масиву і розділяються між собою комами. Допускаються завдання значення індексу у вигляді арифметичного виразу. Максимальна кількість індексів для стандартного Бейсика і Турбо-Бейсика дорівнює восьми, а для Квік-Бейсика - шістдесяті. Значення індексів відраховуються від 0 і не перевершують 32767. Обсяг оперативної пам'яті, що відводиться для збереження всіх рядкових змінних і масивів, використовуваних у програмі, складає 64К. У Квік-Бейсику можливе завдання нестандартного режиму роботи з масивами, при якому окремі масиви можуть займати пам'ять більше 64К.

Оголошення масиву, тобто описання типу, вказівка імені, кількості індексів і їхніх максимальних значень провадиться оператором DIM. У стандартному Бейсику при виконанні цього оператора елементам рядкових масивів як початкове значення присвоюється порожній рядок, а елементам числових масивів – нуль. В усіх версіях при повторному запуску програми провадиться обнулення елементів масивів і змінних.

У програмі може використовуватися масив і без оголошення в операторі DIM. За замовчуванням такий масив вважається одномірним, а максимальне значення індексу дорівнює 10. Якщо в ході рахунка значення індексу стає більшим десяти, то видається повідомлення про помилку.

Оператор оголошення масиву має наступний вигляд:
DIM <ім'я_масиву>(<виміри>) [, <ім'я_масиву> (<виміри>)...]

<виміри> – це список виразів, розділених комами, значення яких повинні бути невід'ємними числами. Ціла частина цих значень визначає верхню границю зміни відповідних індексів.

2. Завдання до лабораторної роботи

1. Розробити програми розв'язку задач (2 завдання) відповідно до варіанту (див. Додаток).

Програма повинна розпочинатися відповідними реквізитами. Наприклад,

Варіант 3. Студент Петренко В.Г.

Значення змінних для обчислення повинні вводитись з клавіатури за допомогою оператора INPUT з певним запрошенням. Наприклад,

Введіть значення a, b, c:

Виведення значень обчислень повинно супроводжуватися відповідним поясненням. Наприклад,

Для a = 7. 8; b = -0. 56; c = 456 найбільше число c = 456

2. Розробити фрагмент конспекту уроку пояснення нового матеріалу щодо обробки масивів та рядкових змінних на мові BASIC з прикладами їх використання.
3. Оформити звіт по лабораторній роботі.

Контрольні питання

1. Які величини називають рядковими?
2. Які операції припустимі над рядковими величинами в мові Бейсик?
3. Як виконується операція з'єднання рядкових величин?
4. Що розуміють під довжиною тексту?
5. Що розуміють під порожнім текстом?
6. Що розуміється під масивом?
7. Як позначаються в програмі елементи масиву?
8. Як організувати введення в пам'ять ЕОМ лінійних масивів?

Додаток

1. Побудувати алгоритм для підрахунку числа додатних елементів в лінійній таблиці.
2. Побудувати алгоритм перепису значень елементів однієї таблиці в іншу.
3. Побудувати алгоритм для підрахунку числа від'ємних елементів в лінійній таблиці.
4. Дана лінійна впорядкована таблиця, що складається з цілих чисел. Побудувати алгоритм, за допомогою якого значення всіх елементів таблиці, менше 10, замінювались би числом 10.
5. Побудувати алгоритм, що визначає скільки раз число 8 зустрічається серед елементів лінійної таблиці A[1:10], яка складається з цілих чисел.
6. Побудувати алгоритм визначення суми всіх чисел, які входять в лінійну таблицю A[1:n].
7. Побудувати алгоритм визначення суми всіх додатних чисел, які входять в лінійну таблицю B[1:n].
8. Побудувати алгоритм визначення номера елемента з максимальним значенням з лінійної таблиці B[1:n] при умові, що значеннями елементів таблиці B є цілі числа.
9. Побудувати алгоритм пошуку мінімального елемента лінійної таблиці A[M:N]. На екран дисплею вивести значення мінімального елемента та його порядковий номер.
10. Побудувати алгоритм, що перевіряє, чи є дане число C менше всіх елементів даної таблиці A[1:n].
11. Скласти алгоритм та програму підрахунку сумарного числа букв "а" та букв "б" в даній літерній величині.
12. Скласти алгоритм заміни в якій-небудь літерній величині всіх букв "а" на букви "б" та навпаки (при такій заміні, наприклад, із слова "баба" повинно одержатись слово "абаб").
13. Скласти алгоритм подвоєння кожної букви в даному тексті (при цьому, наприклад, із слова "бейсик" повинно отриматись слово "ббеейссиик").
14. Скласти алгоритм, який встановлює чи є дане слово "перевертишем". Так називаються слова, які читаються однаково зліва направо та справа наліво. Наприклад, ПОТОП.

15. Скласти алгоритм, який викреслює із даного слова всі букви “а” (так, щоб, наприклад, із слова “заноза” отрималось слово “знос”).

16. Скласти алгоритм, який замінює всі букви “б” на сполучення букв “ку”. Так, щоб, наприклад, із слова “абракадабра” отрималось слово “акуракадакура”.

17. За допомогою операцій вирізки та склеювання складіть із частин слова “інтеграл” слова “гантелі”, “рентген”, “агент”.

18. Написати алгоритм який підраховує скільки раз в даному слові зустрічається сполучення “па” (наприклад, у слові “папараці” воно зустрічається два рази).

19. Написати алгоритм, який підраховує скільки раз в даному слові X зустрічається (як його частина) дане слово Y. Якщо слово Y довше, ніж слово X, то результат повинен бути рівний нулю.

Приклади:

якщо X=“атестат”, Y=“т”, результатом повинно бути число 3;

якщо X=“абракадабра”, Y=“бра”, результатом повинно бути число 2;

якщо X=“забій”, Y=“їж”, результатом повинно бути число 0;

якщо X=“ааааа”, Y=“ааа”, результатом повинно бути число 3.

20. Написати алгоритм, що вказує в даному реченні слово, в якому кількість голосних (а, е, о) максимальна.

ВИКОРИСТАНА ЛІТЕРАТУРА

1. Авербух А.В., Гисин В.Б., Зайдельман Я.П., Лебедев Г.В.. Изучение основ информатики и вычислительной техники: Пособие для учителя. – М.: Просвещение, 1992. –302 с.: ил.
2. Бауэр Ф.Л., Гооз Г. Информатика. - М.: Мир, 1990. -Т. 1, 2.
3. Дибкова Л.М. Информатика та комп'ютерна техніка: Посібник для студ. Вищ. Навч. закл. – К.Б 2002. – 320с.
4. Дидактика средней школы / Под ред. М.Н.Скаткина. - М.: Просвещение, 1982. -320 с.
5. Есаян А.Р. и др. Информатика: Учеб. пособие для педагогических институтов. - М.: Просвещение, 1991. - 288 с.
6. Информатика: Навч. посібн. для 10–11 кл. середн. загальноосвітн. шкіл / І.Т.Зарецька, Б.Г.Колодяжний, А.М.Гуржій, О.Ю.Соколов. – К.: Навчальна книга, 2002. – 496 с.: іл.
7. Кушниренко А.Г., Лебедев Г.В., Сворень Р.А.. Основы информатики и вычислительной техники: Пробный учебник для средних учебных заведений. – 2-е изд. – М.: Просвещение, 1991. –224 с.: ил.
8. Міхеєв В.В. Методика навчання інформатики: Мет. посібник для студ. вищих навч. закл.. – Житомир, 2004. – 224 с.: іл.
9. Міхеєв В.В. Основи інформатики: Методичний посібник для студ. вищих навч. закл.. – Житомир: Вид-во ЖДУ ім. І.Франка, 2005. – 123 с.: іл.
10. Шестопалов Є.А. Информатика, комп'ютерні тести, практичні роботи. Базовий курс. Частина 1. 2004 – 96 с.
11. Шкиль Н.И., Жалдак М.И., Морзе Н.В., Рамский Ю.С. Изучение языков программирования в школе. - К.: Рад. шк., 1989.