



О.В. Костриба
Р.І. Лещук

Інформатика

*Плани-конспекти
уроків
11 клас*





Інформатика. Плани-конспекти уроків 11 клас./ Костриба О.В., Лещук Р.І. — Білогір'я, Хмельницька область, 2007.

Рецензент: *Лещук І.М.* — методист з інформатики Білогірського районного методичного кабінету.

Упорядники:

Костриба О.В. – вчитель інформатики Білогірського НВК.

Лещук Р.І. – вчитель інформатики Хорошівської ЗОШ І-ІІІ ст.

У посібнику подано календарно-тематичне планування та розробки уроків з інформатики відповідно до програми універсального профілю (35 год, 1 год. на тиждень) для 11 класу.

Даний посібник допоможе вчителю підготуватися до уроку та зробити викладання нового матеріалу цікавим і насиченим. Різноманітний добір завдань дозволить оцінити рівень навчальних досягнень учнів.

Посібник може бути корисним вчителям інформатики та для студентів педагогічних вузів, молодих спеціалістів.



ПЕРЕДМОВА

Мета даного видання – надати допомогу вчителю в підготовці та проведенні уроків інформатики в 11 класах загальноосвітніх навчальних закладів.

У посібнику подано календарно-тематичне планування та розробки уроків відповідно до "Програми з інформатики для загальноосвітніх навчальних закладів" (Видавництво "Прем'єр", Запоріжжя, 2003 рік).

Посібник розрахований на вивчення інформатики в 11 класах загальноосвітніх навчальних закладів універсального профілю по 35-годинній програмі (1 година на тиждень). Дані розробки призначені для проведення уроків в комп'ютерному класі, що працює під керівництвом ОС Windows.

Розробки уроків структуровано за загальною схемою: визначено мету, розкрито зміст вивчення матеріалу, вказано форми перевірки засвоєння вивченого матеріалу, наведено вправи, які готують учнів до сприймання нового матеріалу та його осмислення.

Для вивчення нового матеріалу використовуються різноманітні форми та методи (лекції, бесіди, самостійна робота з матеріалом підручника, схем, таблиць, стендів, пам'яток і т.д.). Схеми, запропоновані упорядниками, є стислим конспектом теми, що дозволяє їх використання на різних етапах уроку: при вивченні нового матеріалу, узагальненні і систематизації знань, повторенні та закріпленні набутих знань.

Вправи, які призначені для закріплення матеріалу, передбачають поступове наростання складності, знання теоретичного матеріалу, вміння використати його на практиці та в нестандартній ситуації.

Всі практичні завдання розраховані на 20-25 хв. роботи з комп'ютером і призначені для формування практичних вмінь і навичок та оцінювання навчальних досягнень учнів.

Для контролю навчальних досягнень використовуються різноманітні форми: самостійні роботи, вибіркоче та фронтальне опитування, комп'ютерне тестування тощо.

Пропонований в посібнику матеріал не завжди може бути використаний повністю за відведений час. Учитель на свій розсуд, враховуючи реальні навчальні можливості класу, може підбирати свої та варіювати запропоновані форми роботи.

Ми сподіваємось, що розробки зацікавлять усіх, хто викладає інформатику в 11 класах. На нашу думку, даними розробками зможуть скористатися як досвідчені вчителі, так і початківці.



Календарно-тематичне планування

№	Тема уроку	Дата
Комп'ютерні презентації		
1.	Поняття про комп'ютерні презентації. Знайомство з програмою Microsoft PowerPoint	
2.	Розмітка та оформлення слайдів. Тексти та малюнки в слайдах.	
3.	Імпортування об'єктів, створених в середовищі інших програмних засобів. Вставка діаграм, таблиць, колонтитулів, відеокліпів в слайд. Анімація та звук в слайдах.	
4.	Інтерактивні презентації. Демонстрація презентацій	
5.	Узагальнення і систематизація знань.	
6.	Тематичне оцінювання з теми "Комп'ютерні презентації".	
Табличний процесор		
7.	Поняття про електронні таблиці. Табличний процесор. Основні об'єкти ЕТ.	
8.	Введення та редагування даних. Робота з книгами.	
9.	Форматування електронних таблиць.	
10.	Виконання обчислень в електронних таблицях. Робота з формулами.	
11.	Використання функцій в ЕТ.	
12.	Діаграми в електронних таблицях.	
13.	Упорядкування, фільтрація та пошук даних.	
14.	Виведення табличних даних та діаграм на друк. Узагальнення і систематизація знань.	
15.	Тематичне оцінювання з теми "Табличний процесор".	
Бази даних. Системи правління базами даних		
16.	Бази даних. Системи управління базами даних. Основні об'єкти баз даних.	
17.	Створення баз даних. Таблиці	
18.	Редагування даних. Форми.	
19.	Поля підстановки. Створення зв'язків між таблицями. Ключові поля.	
20.	Сортування, фільтрація і пошук даних.	
21.	Запити	
22.	Звіти	
23.	Тематичне оцінювання з теми "Бази даних. Системи управління базами даних"	
Комп'ютерне моделювання. Основи алгоритмізації та програмування		
24.	Поняття інформаційної моделі. Поняття технології комп'ютерного моделювання. Інформаційні моделі. Побудова	



	моделі. Основні етапи комп'ютерного моделювання.	
25.	Алгоритми. Властивості алгоритмів. Форми подання алгоритму. Виконавець алгоритму.	
26.	Базові структури алгоритмів: слідування, розгалуження, повторення. Графічні схеми базових структур алгоритмів.	
27.	Порядок складання алгоритмів. Поняття програми. Конструювання алгоритмів за методом "зверху донизу".	
28.	Навчальна алгоритмічна мова. Правила опису алгоритмів навчальною алгоритмічною мовою.	
29.	Поняття програми. Мова програмування. Переклад програм з однієї мови на іншу. Поняття про транслятори, систему програмування. Поняття про середовище програмування Turbo Pascal.	
30.	Величини. Основні характеристики величин. Вказівка присвоювання.	
31.	Поняття основного та допоміжного алгоритму. Процедури та функції. Формальні та фактичні параметри. Глобальні та локальні змінні. Розв'язування задач.	
32.	Вказівка розгалуження та вибору, їх опис навчальною алгоритмічною мовою та мовою програмування. Розв'язування задач з використанням вказівки розгалуження.	
33.	Вказівка повторення та її опис навчальною алгоритмічною мовою і мовою програмування. Розв'язування задач з використанням вказівки повторення.	
34.	Табличні величини (масиви) та їх опис мовою програмування. Алгоритми і Програми роботи з табличними величинами.	
35.	Тематичне оцінювання з теми: "Основи алгоритмізації та програмування"	



Урок № 1

<i>Тема:</i>	Поняття про комп'ютерні презентації. Знайомство з програмою Microsoft PowerPoint.
<i>Мета:</i>	Ввести поняття „презентація”, „комп'ютерна презентація”. Формувати уявлення про призначення презентацій та вміння працювати з ними.
<i>Базові поняття й терміни:</i>	Презентація, комп'ютерна презентація, слайд, мультимедіа.

Хід уроку

1. Організаційний етап. Мотивація навчальної діяльності.

Експрес-повторення опорних знань:

- Можливості яких редакторів ми вивчали на уроках? (графічного, текстового)
- Отже, які існують традиційні форми подання інформації? (графічна, текстова)
- Які ще види інформації ви знаєте? (звукова, відео, комбінована)
- Чи можна поєднати різні види інформації? Як ви вважаєте, яким чином це можна зробити? (думки учнів)

Отже, в документах можуть поєднуватись різні форми подання інформації (текстової, графічної, звукової, відео, ефектів анімації, елементів керування тощо). Такі документи називають мультимедійними.

На наступних уроках ми будемо працювати з програмою, яка дозволяє працювати з такими документами. Це програма підготовки презентацій з інтегрованого пакету *Microsoft Office – PowerPoint*.

На вивчення теми "Комп'ютерні презентації" відводиться 6 годин.

Ознайомлення з планом-програмою теми.

Орієнтовний обсяг знань і вмінь (ознайомлення з стендом "Вивчаємо тему").

Під час вивчення даної теми потрібно буде виконати практичні завдання. В кінці теми – тематичне оцінювання - захист власних робіт. (Оцінювання вашої готової презентації).

2. Оголошення теми, мети уроку.

3. Сприйняття і осмислення нового матеріалу.

а) Пояснення нового матеріалу.

Термін "презентація" має два значення. В широкому значенні презентація — це виступ, доклад, захист проекту, тощо. Кажучи узагальнено, представлення ідей, людей, виробів, матеріалів і послуг — це завжди презентація.



Один з напрямків підвищення ефективності презентацій — це використання технічних засобів. Найочевиднішими технічними засобами супроводу презентацій є наочні засоби (плакати з графіками, діаграмами, фотоматеріалами), також можуть використовуватися кіно-, відео- і діапроектори, комп'ютерна техніка, тощо.

У вузькому значенні термін "презентації" це електронний документ особливого роду. Презентація відрізняється від інших додатків пакету Microsoft Office комплексним мультимедійним змістом і особливими можливостями управління відтворенням. Відтворення може бути автоматичним або інтерактивним, у тому числі і дистанційним. Документи даного типу готують за допомогою спеціальних програмних засобів, але при цьому широко використовують і традиційні універсальні засоби, такі як текстові і табличні процесори, графічні редактори, засоби обробки звукової, відеоінформації та інші програми.

Презентація (від англ. presentation – подання, вистава) – це набір картинок-слайдів на певну тему, які зберігаються у файлі спеціального формату. На кожному слайді можна вмещувати довільну текстову, графічну, відеоінформацію, анімації, стереозвук (як синтезований, так і записаний із мікрофона).

Сфера використання презентацій.

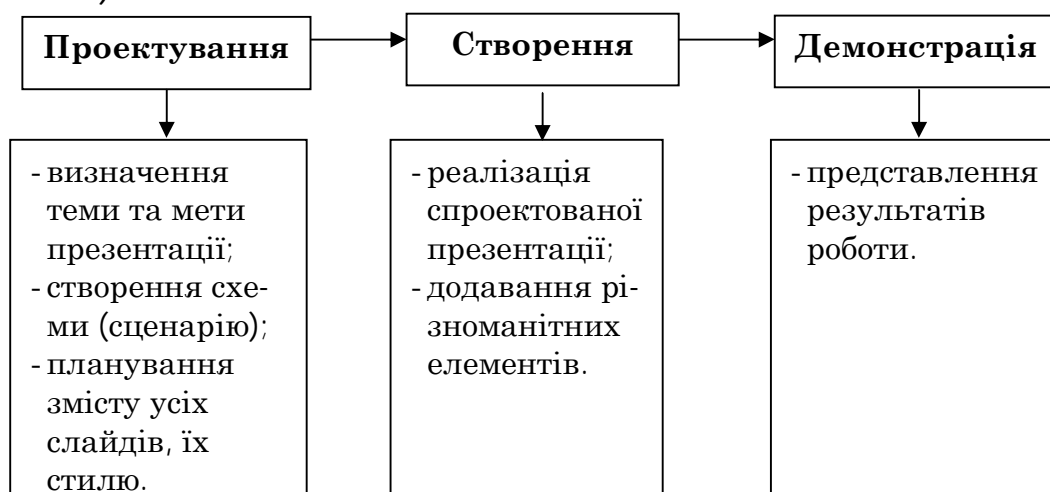
В навчальному процесі для:

- унаочнення навчального матеріалу,
- управління навчально-пізнавальною діяльністю учнів (учні мають змогу опрацювати з матеріал з урахуванням індивідуальних здібностей),
- контролю знань,
- узагальнення та систематизація знань.

В інших сферах діяльності для:

- рекламування товарів, послуг,
- створення фотоальбомів,
- супроводу виступів,
- представлення ідей, тощо.

Робота з презентацією складається з таких етапів (пропонується опорна схема):





За структурою презентації поділяють на *лінійні* та *розгалужені*.

Презентації лінійної структури – послідовне відтворення слайдів.

Презентації *розгалуженої структури* дозволяють змінити порядок відображення слайдів шляхом використання гіпертекстові посилання.

Завантаження програми *Microsoft PowerPoint*:

Пуск – Все програми – Microsoft Office – Microsoft Office PowerPoint.

б) Встановлення зв'язків з матеріалом, вивченим в курсі 10 класу.

Учням слід нагадати, що у всіх програмах Windows операції зі створення, відкриття і зберігання документів виконуються практично однаково.

в) Самостійна робота.

Завдання. Порівняти інтерфейс Word з інтерфейсом PowerPoint, виявити спільні та відмінні елементи, заповнити таблицю)

	Word	PowerPoint
Рядок заголовку	+	+
Головне меню	+	+ нові команди
Робоче поле	Аркуш	слайд
Панелі інструментів	+	+ нові команди

г) Демонстрація учнівських презентацій. (Учні знайомляться з презентаціями, створеними учнями школи, здійснюють аналіз презентацій за критеріями (максимум – 4 бали).

	Зміст	Оформлення	Використання спецефектів (малюнки, анімація)	Кількість балів
Презентація 1				
Презентація 2				
Презентація 3				

д) Колективне обговорення, визначення за підсумками найкращої з представлених презентацій.

е) Ознайомлення з критеріями оцінювання майбутніх учнівських презентацій.

На цьому уроці учні отримують картку для оцінювання, завдяки якій вони можуть орієнтуватися, на що треба звернути увагу при створенні презентації.



Макс. к-сть балів		Само-оцінка	Оцінка вчителя	Оцінка учнів	Середній бал
1 бал	Теоретичний матеріал				
1 бал	Актуальність теми, призначення презентації				
1 бал	Використання малюнків				
1 бал	Використання таблиць, схем, діаграм				
2 бал	Підбір та структура матеріалу, зміст слайдів				
2 бали	Наявність анімаційних ефектів та їх доцільність				
1 бал	Представлення проекту				
1 бал	Відповіді на запитання				
2 бали	Загальне оформлення				
	Підсумок				

4. Усвідомлення набутих знань.

а) *Бесіда*

В ході бесіди з'ясовується, наскільки учні засвоїли поняття „презентація”, „комп'ютерна презентація”.

б) *Самостійна робота учнів з комп'ютерами.*

Інструктаж з техніки безпеки.

Учням пропонується самостійно опрацювати матеріал (додаток 1), щоб дізнатись, якими способами можна створювати презентації і спробувати використати їх на практиці.

Способи створення презентацій.

Додаток 1. Створення презентації.

- *Мастер автосодержания*, який дозволяє вибрати вигляд презентації, задати шаблон оформлення і стилі.
- *Шаблон презентації* — що містить заздалегідь задані параметри розмітки й оформлення презентацій, в які можна вставити свій матеріал.
- Команда *Файл – Создать – Новая презентация* - створення презентації з чистої сторінки.
- Команда *Формат – Разметка слайда* - вибрати розмітку.
- Для першого слайда доцільно вибрати макет *Титульный слайд*.

в) *Обговорення результатів (в групах, потім колективно):*

- Який спосіб найбільше сподобався? Чому?
- Який спосіб дозволяє найкраще проявити творчі здібності?

5. **Домашнє завдання.** Опрацювати додаток 1, 2, конспект уроку.

6. **Підведення підсумків уроку**



Кожен учень, пригадавши тему та мету уроку, виявляє свої досягнення за перший урок.

Додаток 2. Вимоги до структури та змісту навчального матеріалу в презентації.

- Стислий виклад матеріалу, максимальна інформативність тексту.
- Використання знайомих слів і скорочень.
- Ретельно структурована інформація.
- Наявність коротких та лаконічних заголовків, маркованих та нумерованих списків.
- Важливу інформацію (висновки, визначення, правила тощо) треба подавати великим та виділеним шрифтом і розміщувати в лівому верхньому кутку слайда.
- Другорядну інформацію бажано розміщувати внизу слайда.
- Кожному положенню (ідеї) відвести окремий абзац.
- Головну ідею потрібно викласти в першому рядку абзацу.
- Використовувати табличні форми подання інформації (діаграми, схеми) для ілюстрації найважливіших фактів. Це дасть змогу подати матеріал компактно й наочно.
- Пояснення треба розміщувати якнайближче до ілюстрацій, із якими вони мають з'являтися на екрані одночасно.
- Інструкції до виконання завдань необхідно ретельно продумати щодо їх чіткості, лаконічності, однозначності.
- Використовувати емоційний фон.
- Усю текстову інформацію перевірити на відсутність орфографічних, граматичних і стилістичних помилок.
- Продуктивність навчання збільшується, якщо одночасно задіяні зоровий і слуховий канали сприйняття інформації. Там де доцільно, можна використати звуковий супровід для тексту й графічних зображень.



Урок № 2

Тема: Розмітка та оформлення слайдів. Тексти та малюнки в слайдах.

Мета: Формувати вміння проектувати власні презентації, додавати слайди, розміщувати елементи на слайдах та форматувати їх.

Базові поняття й терміни: Структура слайда, розмітка, зміст слайда.

Хід уроку

1. Організаційний етап. Актуалізація опорних знань.

Опитування.

- Назвіть сфери використання презентацій.
- Для чого призначені презентації?
- За допомогою якої програми створюються презентації?
- Яким чином можна завантажити PowerPoint?
- Які є види презентацій (за структурою)?
- Які особливості лінійних презентацій?
- Які особливості розгалужених презентацій?
- Назвіть способи створення презентацій.

2. Оголошення теми, мети уроку.

3. Сприйняття і осмислення нового матеріалу.

а) Самостійна робота учнів з текстом.

Учні самостійно опрацьовують фізіологічні особливості сприйняття кольорів і форм. (Додаток 3).

Додаток 3.

Фізіологічні особливості сприйняття кольорів і форм

- Стимулюючі (теплі) кольори сприяють збудженню й діють як подразники (за спаданням інтенсивності впливу: червоний, оранжевий, жовтий).
- Дезінтегруючі (холодні) кольори заспокоюють, викликають сонливий стан (у тому самому порядку: фіолетовий, синій, блакитний, синьо-зелений, зелений).
- Нейтральні кольори: світло-рожевий, жовто-зелений, коричневий.
- Поєднання двох кольорів — кольору знака й кольору фону — суттєво впливає на зоровий комфорт, причому деякі пари кольорів не тільки стомлюють зір, а й можуть спричинити стрес (наприклад: зелені символи на червоному фоні).
- Найкраще поєднання кольорів шрифту та фону: білий на темно-синьому, чорний на білому, жовтий на синьому.
- Кольорова схема має бути однаковою для всіх слайдів.



- Будь-який малюнок фону підвищує стомлюваність очей і знижує ефективність сприйняття інформації.
- Чіткі, яскраві малюнки, які швидко змінюються, миготять, легко захоплює підсвідомість, і вони краще запам'ятовуються.
- Будь-який другорядний об'єкт, що рухається (анімований), знижує якість сприйняття матеріалу, відвертає увагу, порушує її динаміку.
- Показ слайдів із фоновим супроводженням звуків (пісень, мелодій) викликає швидку втомлюваність, сприяє розсіюванню уваги і знижує продуктивність навчання.

б) Пояснення вчителя по темі "Створення слайдів"

Матеріал уроку доцільно подати в такій послідовності:

1. Структура слайду.
2. Збереження презентацій.
3. Застосування шаблону оформлення слайду.
4. Вставка тексту до слайду.
5. Створення списку.
6. Вставка малюнка до слайду.
7. Демонстрація презентації.

При вивченні матеріалу бажано ілюструвати розповідь вчителя демонстраціями по локальній мережі.

1). Структура слайда

Кожний слайд PowerPoint складається з трьох основних шарів.

- Шар оформлення (рівень основи слайда). Визначає фонове графічне зображення, схему кольорів слайда, шрифти тексту і розташування елементів на слайдах. Шар оформлення єдиний для всіх слайдів презентації.
- Шар розмітки. Визначається типом слайду, що створюється (заголовок, маркований список, діаграма, таблиця або порожній слайд). Розмітка визначає розташування різних елементів на слайді.
- Шар змісту слайда. Призначений для розміщення змісту слайда. Зміст кожного слайда унікальний, тому текст або графічне зображення, що включаються в слайд, з'являються тільки в цьому шарі.

Елементи слайду: заголовки, основний текст, графічні об'єкти (ефекти WordArt, малюнки з бібліотеки ClipArt Gallery (Рисунок, Фильмы и звук) чи прочитані з файлу, автофігури, діаграми, об'єкти WordArt, таблиці), елементи мультимедіа (відеокліпи, звукові кліпи з файлів, дикторські тексти), дата, час, текст колонтитула, номер слайда, кнопки керування процесом демонстрації.

2). Збереження презентацій.

Файл – Сохранить (Сохранить как) – вибрати шлях, де зберігатиметься файл, та ввести ім'я файлу, де зберігатиметься презентація.

**3). Застосування шаблону оформлення слайда.**

Формат - Применить шаблон оформления - вибрати необхідний шаблон оформлення слайда.

4). Вставка об'єктів до слайду.

Вставка тексту WordArt: *Вставка - Рисунок - Объект WordArt* (або на панелі інструментів вибрати кнопку *Добавить объект WordArt*). Вставка тексту: *Вставка - Надпись* (або на панелі інструментів *Рисование* - кнопка *Надпись*).

Текст форматується аналогічно як у текстовому редакторі Word.

5). Створення списку.

- натиснути *Enter* для переходу на новий рядок;
- натиснути кнопку *Нумерация* для створення нумерованого або кнопку *Маркеры* для створення маркованого списку;
- відформатувати список: *Формат - Список... .*

6). Вставка малюнка.

Щоб вставити графічний об'єкт: *Вставка - Рисунок - ...*

7). Демонстрація презентації.

1). В режимі *Обычный* виконуються всі операції.

2). Режим *Сортировщик слайдов* призначений для роботи з презентацією загалом, а не з окремими слайдами. У ньому зручно змінювати порядок розміщення слайдів, переміщувати їх за допомогою миші, підключати різноманітні ефекти при переході від слайда до слайда.

3). У режимі *Показ слайдов* на екран виводять лише слайди підготовленої презентації і кнопки виклику контекстного меню керуванням презентацією.

Режими перегляду презентації

Режим перегляду	Спосіб установки режиму перегляду
Режим слайда (основний режим перегляду слайда при редагуванні)	<i>Вид - Обычный</i> або кнопка вибору режимів перегляду
Режим сортувальника (відображає мініатюрні зображення слайдів, показуючи загальну структуру презентації)	<i>Вид - Сортировщик слайдов</i> або кнопка вибору режимів перегляду
Режим сторінок нотаток (підготовка нотаток доповідача)	<i>Вид - Страницы заметок</i> або кнопка вибору режимів перегляду
Показ слайдів розміром на повний екран	<i>Вид - Показ слайдов</i> або кнопка вибору режимів перегляду
Настроювання параметрів перегляду PowerPoint	<i>Сервис - Параметры</i> - вкладка <i>Вид</i>

Контекстне меню можна також викликати, натиснувши праву кнопку миші у будь-якому місці слайда. Це допоможе під час демонстрації виконувати допоміжні дії. Наприклад, здійснити перехід до будь-якого слайда, викликати примітки до слайда (*Примітки доповідача*).

Потрібний режим задають натисканням на відповідну кнопку в лівій нижній частині екрана або відповідною командою з меню *Вид*.



8). Застосування ефекту прихованого слайда.

Показ слайдов - *Скрити слайд*. Ці слайди не виводяться на екран під час звичайного перегляду презентації. Показати такі слайди на демонстрації можна, вибравши в контекстному меню: *Переход - Вибір слайда по імені* - у запропонованому списку слайдів презентації вибрати прихований слайд. За допомогою цього ефекту можна підготувати деякі додаткові презентаційні матеріали, які можуть бути використані або пропущені під час демонстрації залежно від зацікавленості та реакції аудиторії.

4. Закріплення набутих знань.

Завдання: Спроекувати презентацію, яка б містила інформацію про свій клас. Записати свої ідеї до таблиці.

- Самостійна робота (1 хв.).
- Обговорення в групах по 2 – 3 учні (1 хв.).
- Колективне обговорення.

Заголовок слайда 1: Включає:	Заголовок слайда 2: Включає:	Заголовок слайда 3: Включає:
Заголовок слайда 4: Включає:	Заголовок слайда 5: Включає:	Заголовок слайда 6: Включає:

Створення презентації.

Допомога вчителя	Дії учнів
додавання тексту	вводять коротку інформацію про клас
вставка малюнків (анімації)	відшукують малюнок, який максимально відображає інформацію про клас
кольорове оформлення	використовують оформлення фону та шрифтів
додавання звуку	вставляють до презентації звуковий файл

5. Підведення підсумків уроку

- Проектування презентації.
- Оформлення презентації.
- Вставка об'єктів.

6. Домашнє завдання.

Повторити матеріал "Вимоги до презентацій", "Вставка об'єктів".



Урок № 3

Тема: Імпортування об'єктів, створених в середовищі інших програмних засобів. Вставка діаграм, таблиць, колонтитулів, відеокліпів в слайд. Анімація та звук в слайдах.

Мета: Формувати вміння імпортувати об'єкти, створені в середовищі інших програмних засобів, вставляти діаграми, таблиці, колонтитули в слайд.

Виявити рівень навчальних досягнень з вивченої теми.

Базові поняття й терміни: Імпортування об'єктів, діаграми, таблиці, колонтитули, автофігури, фігурний текст .

Хід уроку

1. Організаційний етап. Актуалізація опорних знань.

Самостійна робота.

Скласти алгоритм ...

- збереження презентацій;
- застосування шаблону оформлення слайда;
- вставки тексту до слайду;
- створення списку;
- вставки малюнка до слайду;
- демонстрації презентації.

2. Оголошення теми уроку.

3. Сприйняття і осмислення нового матеріалу.

Пояснення вчителя.

Імпортування чи вставляння об'єктів можна виконати таким чином:

1) Вставка організаційної діаграми (схеми): *Вставка - Рисунок - Організаційна діаграма.*

2) Вставка та форматування таблиці: *Вставка - Таблиця.* На екрані з'явиться діалогове вікно *Вставка таблиці*, в якому необхідно встановити кількість стовпчиків та кількість рядків у таблиці. Після цього повинна з'явитись таблиця та панель інструментів *Таблиць і границь* (якщо її немає, то виконати: *Вид - Панелі інструментів - Таблиць і границь*).

За допомогою кнопок панелі інструментів можна виконати необхідне форматування: тип, товщину та колір ліній, тип та колір меж, заливку, об'єднати чи поділити комірки, вставити чи вилучити стовпчики чи рядки тощо.

3) Імпортування об'єктів, створених у середовищі інших програмних засобів: *Вставка - Об'єкт* - вибрати тип об'єкта (креслення, документи і



малюнки, сторінки Word чи діаграми Excel тощо) та програму, в якій він виконаний.

4) *Вставка* колонтитулів у слайд: *Вид – Колонтитулы* (нумерація слайдів, вставка дати і часу, іншої текстової інформації) Слід вказати, це стосується одного слайда чи всіх.

4. Засвоєння та закріплення набутих знань.

а) *Розгляд таблиць*. Використання графіки та об'єктів. Конструювання презентацій.

Дія	Способи виконання
Вибір розмітки слайдів	<i>Формат - Разметка слайда</i> Кнопка <i>Разметка слайда</i> на панелі інструментів <i>Стандартная</i>
Застосування оформлення слайдів	<i>Формат - Оформление слайду</i> Кнопка <i>Конструктор</i> на панелі інструментів <i>Форматирование</i>
Створення напису	<i>Вставка - Надпись</i>
Виконання нескладних малюнків	За допомогою панелі інструментів <i>Рисование</i> (кнопка панелі інструментів <i>Стандартная</i>)
Вставка малюнків із файлу	<i>Вставка - Рисунок - ... из файла</i>
Вставка малюнків, фотографій, звуків, відеороликів із колекції ClipGallery	<i>Вставка - Рисунок - Картинки</i> <i>Вставка - Фильмы и звук - ... из файла</i>
Вставка автофігур	<i>Вставка - Рисунок - Автофигуры</i>
Вставка фігурного тексту	<i>Вставка - Рисунок - Объект WordArt</i>
Вставка діаграми	<i>Вставка - Диаграмма - Microsoft Graph</i>
Вставка схематичної діаграми	<i>Вставка - Схематична диаграмма</i>

Конструювання презентацій

Дія	Способи виконання
Настроювання переходу слайдів	<i>Показ слайдов - Смена слайдов</i>
Застосування ефектів анімації до об'єктів слайда	<i>Показ слайдов - Эффекты анимации</i> Кнопка <i>Эффекты анимации</i> на панелі інструментів <i>Форматирование</i>
Додавання ефектів анімації до об'єктів слайда	<i>Показ слайдов - Настройка анимации</i> Кнопка <i>Эффекты анимации</i> панелі інструментів <i>Форматирование</i>
Перегляд ефектів анімації	<i>Область задач - Дизайн слайда</i> - кнопка <i>Перегляд</i>
Додавання відеоролика з колекції	<i>Вставка - Фильмы и звук - Фильм из коллекции картинок</i>
Додавання відеоролика з файлу	<i>Вставка - Фильмы и звук - Фильм из файла</i>
Додавання звукового супроводу з колекції	<i>Вставка - Фильмы и звук - Звук из коллекции</i>
Додавання звукового супроводу з файлу	<i>Вставка - Фильмы и звук - Звук из файла</i>



Додавання звукового супроводу з компакт-диска	Вставка - Фільми и звук - Запись с компакт-диска
Запис звукового супроводу з мікрофона	Вставка - Фільми и звук - Записать звук

б) Практичне завдання. Інструктаж з техніки безпеки.

- Створити презентацію "Мій клас" на 3 слайди.
- Вставити в титульний слайд малюнок.
- Вставити в 2 слайд таблицю "Успішність учнів мого класу" (заповнену).

№	Прізвище, ініціали	Алгебра	Геометрія	Фізика

- Вставити в 3 слайд діаграму "Успішність учнів мого класу" .
- Продемонструвати презентацію, зберегти її у вказаній папці, закрити вікно презентації.

5. Підбиття підсумків уроку. Оцінювання.

а) Самооцінка учнів.

б) Оцінка вчителя (учні, які раніше виконали завдання практичної роботи, працюють над власною презентацією).

6. Домашнє завдання.

Опрацювати конспект теми, таблиці - додатки. Підготувати повідомлення про використання гіперпосилань (2 учні) та кнопок керування (2 учні). Обрати тему для власної презентації та підібрати матеріал для неї.



Урок № 4

Тема: Інтерактивні презентації. Демонстрація презентацій.

Мета: Формувати вміння демонструвати презентації, додавати гіперпосилання, кнопки керування, анімаційні ефекти, звуки.

Базові поняття й терміни: Інтерактивна презентація, кнопки керування, гіперпосилання.

Хід уроку

1. Перевірка домашнього завдання.

Учні по черзі представляють тему презентації. Іншим учням пропонується відповісти на запитання:

- Чи сподобалась вам тема роботи?

2. Оголошення теми уроку.

3. Вивчення нового матеріалу.

а) Дослідницька робота в групах. Інструктаж з техніки безпеки. "Розгляд анімаційних ефектів".

Використовуючи таблицю та спираючись на власний досвід, кожній групі пропонується дослідити по декілька анімаційних ефектів, що використовуються для переходу слайдів, виявити їх властивості та продемонструвати результати роботи учням з інших груп. Для роботи використовується презентація "Мій клас".

I група – жалюзі, прямокутник, шашки.

II група – об'єднання, наплив, прорізання.

III група – відкриття, кути, годинникові стрілки.

<i>Ефект</i> \ <i>Властивість</i>	<i>Швидкість</i>	<i>Звук</i>	<i>Зміна слайда</i>

- звіт груп;
- запитання.

☞ Використання великої кількості ефектів, знижує якість сприйняття презентації.

б) Вивчення нового матеріалу.

Використання гіперпосилань

а) Заслуховується повідомлення учня.

б) Завдання учням: із слайда, що містить список учнів, створити гіперпосилання на слайд з інформацією про конкретного учня.

☞ Слід бути обережними при одночасному використанні гіперпосилань та анімаційних ефектів зміни слайдів.



Кнопки керування

- а) Заслуховування повідомлення учня.
- б) Завдання учням: із слайду, що містить інформацію про конкретного учня, забезпечити перехід на слайд із списком учнів за допомогою кнопки керування.

4. Засвоєння та закріплення набутих знань.

Практичне завдання. Створення презентацій. Демонстрація презентацій. *Інструктаж з техніки безпеки.*

Учні працюють самостійно над створенням власних презентацій, при потребі отримують консультацію вчителя.

5. Підбиття підсумків уроку.

6. Домашнє завдання. Опрацювати конспект уроку.



Урок № 5

Тема:	Узагальнення і систематизація знань.
Мета:	Узагальнити і систематизувати знання учнів по темі "Презентації". Продовжити формування практичних вмінь і навичок щодо створення презентацій. Здійснити поточний контроль знань учнів з теми.

Хід уроку

1. Організаційний етап.

2. Оголошення теми уроку.

3. Контроль і корекція набутих знань, вмінь, навичок.

а) Комп'ютерне тестування. Інструктаж з техніки безпеки.

(Використовується контрольна-діагностична програма Test-W2 або аналогічна, запитання тесту підготовлюються вчителем).

Орієнтовні запитання:

1. Як додати малюнок до слайду?
2. Як можна змінити порядок демонстрації слайдів у презентації?
3. Яке призначення режиму демонстрації?
4. Як вставити новий слайд?
5. Що таке форматування елементів слайдів?
6. В якому режимі можна редагувати слайд?
7. Які способи існують для створення презентацій?
8. Слайд може містити: ...
9. Основний елемент презентації - ...
10. Програма PowerPoint призначена для ...
11. Переміщення об'єктів на екрані під час демонстрації - це ...
12. Основний елемент презентації ...
13. Що таке анімація?

б) Практичне завдання. Інструктаж з техніки безпеки.

- Завершення роботи над презентацією "Мій клас".
- Демонстрація презентації "Мій клас".
- Оцінювання робіт згідно критеріїв.

4. Формування практичних вмінь і навичок.

Практичне завдання. Інструктаж з техніки безпеки.

Індивідуальна робота над власною презентацією. Консультування окремих учнів (за потреби).

5. Домашнє завдання.

Повторити конспекти уроків та роздані пам'ятки по темі "Презентації".
Продумати захист власної презентації.

6. Підведення підсумків уроку.



Урок № 6

<i>Тема:</i>	Тематичне оцінювання з теми "Комп'ютерні презентації".
<i>Мета:</i>	Здійснити тематичний контроль знань та вмінь учнів з теми „Комп'ютерні презентації”. Розвивати уяву та творчі здібності учнів.

Хід уроку

1. Організаційний етап.
2. Оголошення теми уроку.
3. Тематичне оцінювання.

а) Теоретична частина.

Тестування проводиться за допомогою програми Test-W або аналогічної програми. (Тести по темі "Презентації").

б) Робота над презентаціями. Інструктаж з техніки безпеки.

1) Робота в парах.

Обговорення та перегляд власної презентації (при виконанні цього завдання учні діляться зі своїми однокласниками своїми досягненнями, учень-слухач може висловити свої ідеї, пропозиції).

2) Самостійна робота учнів

- Учні завершують роботу над презентаціями.
- Підготовка до захисту.
- Рекомендації вчителя щодо захисту проектів.
- Визначається черговість виступів учнів (жеребкування).
- Захист презентацій.

Учні захищають власні роботи. На захист запрошені учні з іншої підгрупи цього ж класу для допомоги в оцінюванні та підрахунку кількості балів та відповідної оцінки.

- Оголошення оцінок.

4. Підведення підсумків уроку і теми в цілому.

Учні мають можливість висловити свої думки:

- чи сподобались форми роботи, використані при вивченні теми?
- чи задовольняють результати власної діяльності?
- побажання іншим учням, вчителю;
- відношення до створення презентацій та використання їх в навчально-виховному процесі.

5. Домашнє завдання:

- продумати, що можна змінити у презентації;
- повторити види прикладного програмного забезпечення;
- підготувати повідомлення та презентацію про призначення електронних таблиць (2–3 учні).



Урок № 7

<i>Тема:</i>	Поняття про електронні таблиці. Табличний процесор. Основні об'єкти ЕТ.
<i>Мета:</i>	Розглянути поняття "електронні таблиці", "табличний процесор", системи опрацювання табличної інформації, основні об'єкти ЕТ, структуру вікна MS-Excel. Вивчити призначення та основні функції табличного процесора. Навчити учнів завантажувати табличний процесор. Ознайомити з галузями використання MS-Excel.
<i>Базові поняття й терміни:</i>	Електронні таблиці, табличний процесор, робоча книга, робочий лист, комірка, стрічка, стовпчик, адреса комірки, поле імені.

Хід уроку

1. Організаційний етап

Ви починаєте вивчати нову тему – "Табличний процесор". На дану тему відводиться 10 годин (уроків), по закінченню – тематичне оцінювання у вигляді практичної роботи. Ця тема є частиною розділу "ППЗ загального призначення".

На уроках ми будемо вивчати теоретичний матеріал, виконувати різноманітні практичні завдання.

В результаті ви повинні отримати певні знання, вміння та навички. Учням потрібно пояснити, де вони можуть отримати інформацію про вимоги до своєї діяльності. (Стенди на зразок "Вивчаємо тему", посібники "Учні повинні знати та вміти", "Критерії оцінювання").

2. Мотивація навчальної діяльності учнів

Одним з найважливіших засобів опрацювання і зберігання інформації є таблиці. Вони можуть утворювати документ або бути його частиною.

Програми, які забезпечують автоматизацію процесу створення, опрацювання, корегування, зберігання і виведення на друк документів у формі таблиць або їх частин, надзвичайно корисні і потрібні, і ось такій програмі будуть присвячені уроки нашої теми.

3. Оголошення теми і мети уроку.

4. Вивчення нового матеріалу.

Розповідь вчителя з використанням демонстрації по локальній мережі.



Електронна таблиця – сукупність рядків та стовпчиків, стовпчики позначені літерами латинського алфавіту, а рядки цифрами. Перетин рядка і стовпчика називається коміркою. Кожна комірка (клітинка) має свою адресу, яка складається з літери стовпчика та цифри рядка (А5, В7, К4,...). Рядки: 1...65536. стовпці: А-ІV (всього 256).

Сучасні електронні таблиці володіють власними засобами програмування, дозволяють здійснювати публікацію в Web, володіють засобами захисту даних.

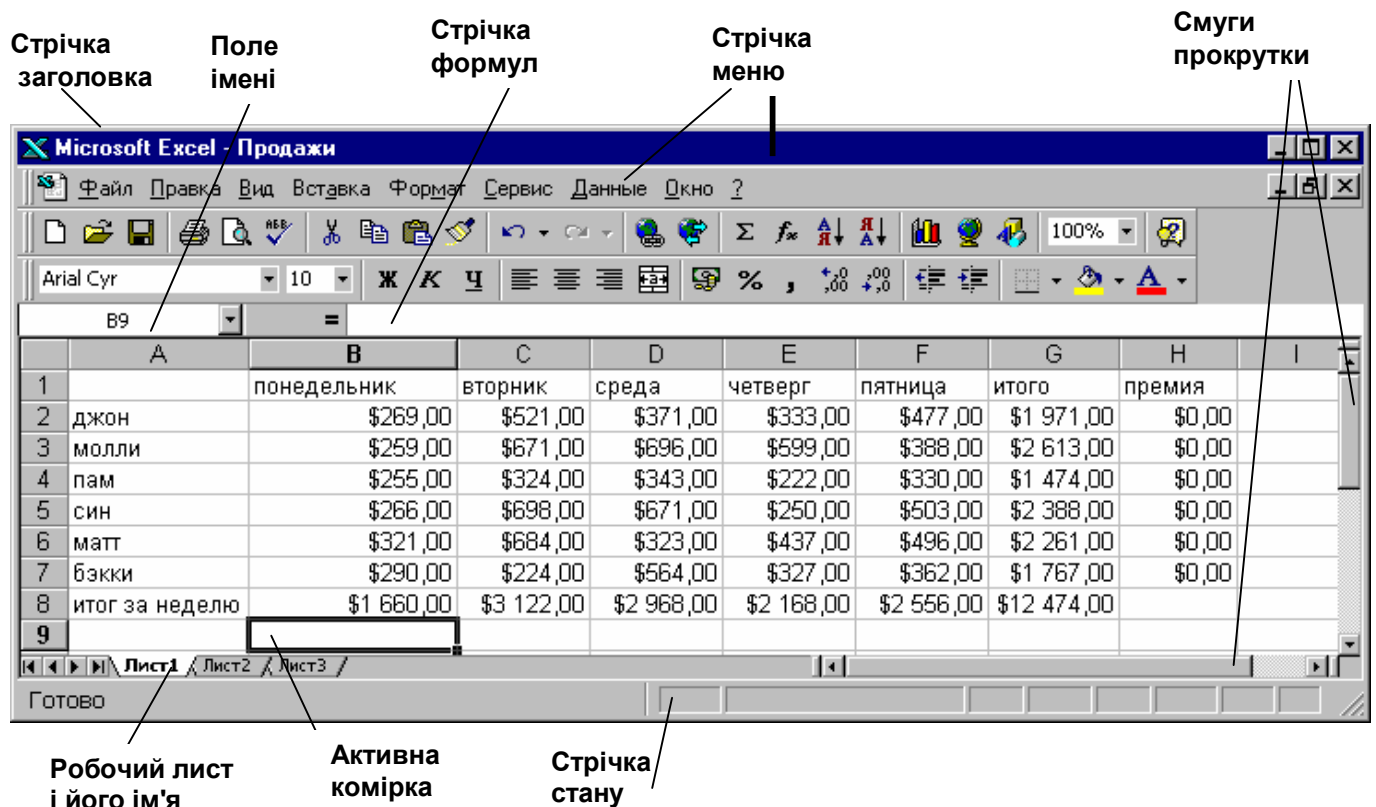
У пам'яті комп'ютерів можна зберігати електронні таблиці великого розміру. На екрані можна утворювати кілька вікон з різними частинами таблиці.

Електронна таблиця - це комп'ютерна програма, яка забезпечує автоматизацію процесу створення, опрацювання, корегування, зберігання і виведення на друк документів у формі таблиць або їх частин. Часто багатofункціональні ЕТ називають табличним процесором.

Приклади ЕТ: SuperCalc, Quattro Pro, Lotus, Microsoft Excel та ін.

Порівняно простими в використанні та поширеними є електронні таблиці Microsoft Excel. І на наших уроках ми вивчатимемо електронні таблиці на прикладі саме Microsoft Excel, що входить до складу пакету Microsoft Office.

Інтерфейс Програми



Елементи (об'єкти) головного вікна Excel.

Робоча книга. Файл Excel, який містить декілька робочих листів.



Робочий лист. Область в Excel, яка складається з комірок. Кожна робоча книга містить в собі 3 листа. Ми можемо змінювати їх кількість.

Комірка. Одна з прямокутних рамок, з яких складається робочий лист. В комірці можна вводити будь-яку інформацію: числа, текст, математичні формули, логотипи і складні графічні зображення.

Стрічка. Група комірок, які розташовані в один ряд по горизонталі. Кожна стрічка має свій номер.

Стовпчик. Група комірок, які розташовані в один рядок по вертикалі. Кожний стовпчик позначається однією чи декількома буквами.

Адреса комірки. Ідентифікатор, який однозначно визначає комірку шляхом вказування букви, яка задає стовпчик, і порядкового номеру стрічки.

Ім'я робочого листа. Ім'я кожного робочого листа відображається на ярличку, який розташований в нижній частині вікна Excel.

Стрічка заголовка програми містить ім'я програми. В правому верхньому кутку екрану знаходяться два ряди кнопок. Кнопки верхнього ряду дозволяють змінювати розміри вікна самої Excel, а нижнього – її робочих книг.

Стрічка меню. Саме з неї починається виконання багатьох завдань в Excel.

Поле імені і стрічка формул. В полі імені відображається ім'я активної комірки. Стрічка формул використовується для зміни даних, які знаходяться в активній комірці.

Смуги прокрутки. Дозволяють проглядати ті частини робочого листа, які не видно на екрані.

Стрічка стану. Відображає різні повідомлення і поточний стан робочої книги.

Після запуску програми автоматично створюється документ Книга1. Інтерфейс Excel багатодокументний. Кожний документ в Excel називається Робочою книгою. Файли Робочих книг мають розширення .xls, а шаблонів документів .xlt.

Кожна книга складається з листів (аркушів). По замовчуванню книга містить три листи (Лист1, Лист2, Лист3). Той лист, який в даний момент відкритий називається активним.

Виділена комірка називається активною, або поточною. Адреса активної комірки відображається в полі імені.

Діапазон стовпців – це вертикальна смуга таблиці. Діапазон В:Е – це вертикальна смуга шириною 4 стовпці від стовпця з ім'ям В до стовпця з ім'ям Е включно.

Діапазон рядків – це горизонтальна смуга таблиці. Діапазон 3:8 – це горизонтальна смуга шириною в 6 рядків від 3 до 8 включно.

Блок комірок – це прямокутник, що задається адресою лівої верхньої і правої нижньої комірок. Наприклад блок В3:Е8 знаходиться на пере-



тині вертикальної смуги В:Е і горизонтальної смуги 3:8, він містить $4 \times 6 = 24$ комірки.

Властивості комірки:

- Тип та колір фону.
- Тип, розмір та колір шрифту.
- Тип, колір і місце проведення границі.
- Тип вирівнювання.
- Тип орієнтації тексту.
- Тип даних.
- Захист.

Можливості програми:

- Введення і редагування даних.
- Форматування комірок, рядків та стовпчиків таблиці.
- Введення формул (автоматизація розрахунків).
- Використання широкого спектру різних функцій.
- Побудова, редагування та друк діаграм.
- Попередній перегляд та друк таблиці.
- Створення та введення баз даних.

Отже, великий попит на використання електронних таблиць пов'язаний з їх універсальністю, адже без обчислень не можна обійтись у багатьох сферах нашого життя.

Табличні процесори призначені для виконання обчислень, створення діаграм, звітів, розв'язування складних задач у сфері природничих, технічних наук та ін.

5. Закріплення вивченого.

Експрес - опитування учнів:

- Що називається електронною таблицею?
- Для чого створені електронні таблиці?
- Перелічіть основні сфери застосування електронних таблиць.
- Які можливості електронних таблиць ви знаєте?
- Які вам відомі елементи (об'єкти) головного вікна Excel?
- З якою програмою схожий інтерфейс програми Excel?

6. Домашнє завдання. Опрацювати конспект уроку.

Р.1, §1,2, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов, § 12.1 "Курс інформатики". В.Д. Руденко, О.М. Макаруч, М.О. Патланжоглу.

7. Підсумок уроку.



Урок № 8

Тема: Введення та редагування даних. Робота з книгами.

Мета: Розглянути головні способи та прийоми введення і редагування даних в ЕТ.
Навчити учнів вводити текстові та числові дані до таблиці, виконувати операції з файлами, в яких зберігаються електронні таблиці, працювати з листами книги.

Базові поняття й терміни: Типи даних, стрічка формул, стрічка стану.

Хід уроку

1. Мотивація навчальної діяльності учнів.

На минулому уроці ми вивчали теоретичний матеріал про електронні таблиці та табличні процесори. Сьогодні ми будемо вчитись на практиці самостійно вносити та редагувати табличні дані, створювати готові документи (книги), працювати з сторінками (листами).

2. Оголошення теми і мети уроку.

3. Відтворення і корекція опорних знань учнів

Проводиться опитування, в результаті якого учні повинні дати означення понять, що були розглянуті на минулому уроці, зокрема таких як *електронні таблиці, табличний процесор, робоча книга, робочий лист, комірка, стрічка, стовпчик, адреса комірки, поле імені*.

Далі у формі бесіди з'ясовується можливості електронних таблиць, і, як висновок, їх призначення та необхідність.

3. Вивчення нового матеріалу.

Пояснення вчителя з використанням демонстрації по локальній мережі.

Введення даних в ЕТ.

Дані у ЕТ бувають різних типів, наприклад: число, текст, формула.

Вводити дані можна безпосередньо до комірки, а можна і в рядок формул. Дані вводяться в основному з клавіатури або з використанням інших методів (вставка готових, автозаповнення, перетягування, та ін.)

Для того, щоб активізувати комірку, необхідно помістити курсор на ній (активізована комірка буде обрамлена чорним кольором). Після завершення введення даних у комірку необхідно натиснути Enter або скористатись клавішами керування курсором.

Елементарні способи редагування таблиць:

- зміна даних (переміщення і копіювання даних, перетягування і кидання, використання буфера обміну, маркер заповнення);
- вставка рядків і стовпців;



- видалення рядків, стовпців і комірок;

Функції копіювання, вирізання, перенесення у інше місце ті ж самі, що і у Microsoft Word та Microsoft Power Point.

Для виділення суміжного діапазону курсор поміщають у правий верхній його кут і виділяють, протягуючи курсор по діагоналі вниз. Для виділення несуміжного діапазону, використовують клавішу Ctrl.

Копіювання, вирізку та очистку виділеного діапазону проводять стандартними командами або комбінаціями клавіш.

Щоб вставити у потрібне місце скопійований чи вирізаний діапазон, поміщають курсор у комірку, яка буде правим верхнім кутком цього діапазону і натискають на відповідну піктограму. Але, коли потрібно вставити комірки, обчислені за формулами, і які посилаються на попередні комірки, то при копіюванні адреси можуть змінитися, а тому, замість піктограми, якою позначається вставка, виконують команду *Правка - Спеціальна Вставка* і вибирають тип даних, який потрібно перенести - числа чи формули.

Для того, щоб вставити порожній рядок (стовпчик), потрібно активізувати будь-яку комірку рядка (стовпчика), перед яким потрібно вставити порожній, і виконати команду *Вставка – Строка (Столбец)*.

Для того, щоб вилучити рядок (стовпчик), активізують будь-яку комірку цього рядка (стовпчика) і виконують команду *Правка - Удалить - Строка (Столбец)*.

☞ **Якщо почати вводити нові дані у клітинку, то старі зникають!**


Для редагування вмісту клітинки можна використати методи:

- поставити курсор на потрібну клітинку і натиснути клавішу F2, потім з допомогою клавіш керування курсором, клавіш Delete та BackSpace змінити написане;

- двічі швидко клацнути по комірці, потім змінити написане;

- підвести вказівку миші до рядка формул і клацнути в ньому, в самому рядку формул зробити зміни.

Всі зміни підтверджуються клавішею Enter.

Для одночасного виконання об'єднання та вирівнювання по центру існує кнопка на панелі інструментів –  — "об'єднати і помістити в центрі".

Як зазначалось на минулому уроці, інтерфейс Excel багатодокументний. Кожний документ в Excel називається *Робочою книгою*. Файли *Робочих книг* мають розширення .xls, а шаблонів документів .xlt.

Збереження робочих листів.

Для збереження нового файлу використовується діалогове вікно збереження документу:

- Виберіть *Файл - Сохранить*. На екран буде виведене діалогове вікно *Сохранение* документа. Введіть в текстовому полі ім'я файлу.



- Клацніть на кнопці *Сохранить*.
- Якщо потрібно *Сохранить* старий файл під новим іменем, необхідно вибрати *Файл - Сохранить как...*

4. Закріплення вивченого.

Практичне завдання в табличному процесорі Excel.

Пам'ятка:

- Для переміщення по книгах та листах книги потрібно використовувати мишу та клавіші управління курсором.
- Для введення інформації в комірку потрібно помістити курсор в комірку та ввести дані в стрічці введення даних.
- Для редагування даних використовуються клавіші F2, Del, Ins, Enter та інші.
- Під час роботи потрібно періодично зберігати файл.
- Для роботи з виділеними даними при необхідності використовуються пункти меню *Правка*.

5. Практичне завдання. Інструктаж з техніки безпеки.

1. На першому листі робочої книги створити таблицю за зразком:

Відомість обігу товарів

Найменування товару	Ціна грн	Вхідний залишок		Надходження		Видаток	
		К-ть, кг	Сума, грн	К-ть, кг	Сума, грн	К-ть, кг	Сума, грн
Ковбаса	12,4	68,5		23,3		43,8	
Балик	19,5	35,7		18,7		21,5	
М'ясо	6,2	29,3		31,5		41,3	
Печінка	5,6	41,1		12,6		29,2	
Сметана	2,1	21,5		10,7		15,2	
Молоко	1,0	49,2		14,2		32,3	
Усього по складу							

2. Зберегти книгу з таблицею під назвою "Товари".

6. Домашнє завдання. Опрацювати конспект уроку.

Р.1, §3, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов, § 12.2, 12.3, 12.5 "Курс інформатики". В.Д. Руденко, О.М. Макарчук, М.О. Патланжоглу.

7. Підсумок уроку.



**Урок № 9**

Тема:	Форматування електронних таблиць.
Мета:	Закріпити знання учнів про редагування даних та роботу з книгами, з аркушами. Ввести поняття "формат комірок", ознайомити з правилами розміщення даних в електронних таблицях. Навчити задавати коміркам різні типи даних.
Базові поняття й терміни:	Формат, форматування,

Хід уроку

1. Організаційний момент**2. Відтворення і корекція опорних знань учнів**

З допомогою учнів учитель заповнює праву частину заздалегідь підготовленої на дошці таблиці:

Дія	Способи виконання
Початок редагування комірки	Клацання мишею в рядку введення даних
	Подвійне клацання в потрібній комірці
	Натискання клавіші "F2"
Завершення редагування із збереженням змін	Клацання мишею по кнопці  в рядку формули
	Перехід на іншу комірку
	Натискання клавіші "Enter"
Завершення редагування без збереження змін	Клацання мишею по кнопці  в рядку формули
	Натискання клавіші "Esc"

3. Оголошення теми і мети уроку.**4. Вивчення нового матеріалу.****Розповідь вчителя.**

Форматування електронних таблиць найчастіше включає в себе такі операції:

- зміна формату даних;
- форматуванням вмісту комірок;
- форматуванням рядків і стовпців;
- використання стилів;

Формат даних – можливі варіанти: числа, логічні значення, текст, дата і час.



Форматуванням вмісту комірки називають встановлення вигляду відображення вмісту комірки. Для форматування комірки або виділеного діапазону комірок використовують команду *Формат* або кнопки панелі інструментів *Форматирование*.



- Тип та колір фону.
- Тип, розмір та колір шрифту.
- Тип, колір і місце проведення границі.
- Тип вирівнювання.
- Тип орієнтації тексту.
- Числовий тип.
- Захист.

Поняття формату даних комірки включає такі параметри: шрифт (тип, розмір, накреслення, колір); формат чисел; спосіб вирівнювання; розміри (ширина і висота) комірок; обрамлення комірок; межі та їх параметри.

Форматування можна здійснювати за допомогою вкладок автоформат, формат чисел, вирівнювання даних у комірках, розміри колонок і рядків, оформлення виділеного діапазону, стиль форматування.

Якщо потрібно зробити зміни у форматі таблиці чи окремих комірок, які не можна виконати за допомогою панелі інструментів, то у пункті меню *Формат* вибирають необхідні розділи.

Наприклад, якщо потрібно внести якісь зміни у форматі комірок (змінити тип, наприклад, з текстового на числовий, змінити розміри - задати точну ширину стовпців, висоту рядків та ін.), користуються підпунктом меню *Формат – Ячейки* (з'явиться діалогове вікно *Формат ячеек*). Якщо потрібно оформити всю таблицю за певним форматом, її виділяють, і, виконавши команду *Формат – Автоформат*, вибирають потрібний.

Досить часто необхідно змінити розміри стовпця чи рядка. Це робиться досить просто: ви підводите вказівку миші на межу стовпців (рядків) до появи подвійної стрілочки $\left\langle \right\rangle$, натискаєте на ліву кнопку миші, і не відпускаючи її, тягнете в потрібну сторону.

Також є можливість точного задання висоти однієї або багатьох стрічок. Найлегше це можна зробити з контекстного меню. Аналогічно – стовпці.

5. Практичне завдання. Інструктаж з техніки безпеки.

1. Відкрити створену на попередньому уроці книгу "Товари"



2. Відформатувати таблицю "Відомість обігу товарів" за зразком. (Додалось дві колонки!)
3. Прозвітувати про виконання роботи вчителю.
4. Зберегти результати роботи і закрити всі вікна.

Відомість обігу товарів

Найменування товару	Ціна грн	Вхідний залишок		Надходження		Видаток		Залишок на кінець місяця	
		К-ть, кг	Сума, грн	К-ть, кг	Сума, грн	К-ть, кг	Сума, грн	К-ть, кг	Сума, грн
Ковбаса	12,4	68,5		23,3		43,8			
Балик	19,5	35,7		18,7		21,5			
М'ясо	6,2	29,3		31,5		41,3			
Печінка	5,6	41,1		12,6		29,2			
Сметана	2,1	21,5		10,7		15,2			
Молоко	1,0	49,2		14,2		32,3			
Усього по складу									

6. Підведення підсумків роботи

При необхідності обговорюються проблеми та помилки, що виникали під час роботи.

7. Закріплення вивченого матеріалу.

- За допомогою яких клавіш можна редагувати дані в таблиці?
- За допомогою чого можна вставляти дані в комірки формули?
- Як можна змінити параметри таблиці (ширину, висоту стовпців та стрічок)?
- За допомогою яких команд можна змінювати кількість стовпців, стрічок, комірок таблиці?
- Як можна видалити за один раз цілий стовбець або рядок?
- Як можна знищити за один раз цілу таблицю?

8. Домашнє завдання: Опрацювати конспект уроку.

Р.1, §4. Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов.

§ 12.6 "Курс інформатики". В.Д. Руденко, О.М. Макарчук, М.О. Патланжоглу.

9. Підсумок уроку.



Урок № 10

Тема: Виконання обчислень в електронних таблицях. Робота з формулами.

Мета: Сформувати поняття про можливі способи виконання обчислень в електронних таблицях. Дати чітке уявлення про пріоритет операцій. Пояснити різницю між абсолютними та відносними посиланнями. Навчити вводити формули до таблиці, виконувати копіювання формул. Вдосконалити навички роботи з аркушами.

Базові поняття й терміни: Формули, оператори, пріоритет операцій, абсолютні та відносні посилання

Хід уроку

1. Організаційний етап
2. Оголошення теми і мети уроку.
3. Вивчення нового матеріалу.

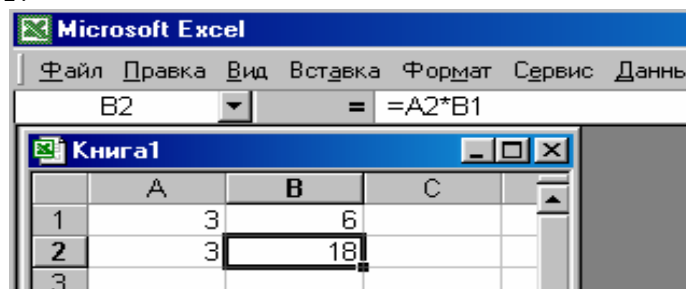
Розповідь вчителя з використанням демонстрації по локальній мережі.

Найпростіші обчислення можна виконати, почавши з знака "=", і записавши математичний вираз. Приклад: $=40+5*2$. Результат – в клітці вивітється число 50.

Обчислення в середовищі електронних таблиць Excel здійснюються за допомогою формул. Формула може містити числові константи, посилання на клітинки і функції, з'єднані знаками математичних операцій. Дужки дозволяють змінювати стандартний порядок виконання дій. Якщо зробити клітинку активною, то сама формула відображається в рядку формул.

Формула в електронній таблиці – це сукупність арифметичних операцій, адрес клітинок та звернень до функцій.

Введення формули починається зі знаку "=". Якщо в клітину введена формула, то в клітинці після натиснення клавіші введення буде видно результат, а в рядку введення саму формулу, за допомогою якої був отриманий результат.



Знаки арифметичних операцій:

+ - додавання



- - віднімання
- x - множення
- / - ділення
- ^ - піднесення до степеня
- () - підвищення пріоритету операції

Особливості запису формул.

Для введення формули курсор поставити на потрібну комірку.

1. Запис адреси комірок відбувається англійською мовою.
2. Запис формул починається із знака "=".
3. Десяткові дробі записуються з комою (3,5).

☞ **Порада:** користуйтеся додатковою цифровою клавіатурою – вона містить всі знаки арифметичних дій, крім дужок і знаку "=" (При користуванні повинен горіти індикатор Num Lock.)

Формула може містити посилання, тобто адреси клітинок, вміст яких використовується в обчисленнях. Це означає, що результат обчислення формули залежить від числа, що знаходиться в іншій клітинці. Посилання можна задавати різними способами. По-перше, адресу клітинки можна ввести вручну; по-друге — можна клацнути мишкою на відповідній клітинці чи вибрати діапазон клітинок, адресу якого потрібно ввести. Для редагування формули слід двічі клацнути мишею на відповідній клітинці. При цьому клітинки (діапазони), від яких залежить значення формули, виділяються на робочому листі рамками, а посилання відображаються в клітинці і в рядку формул тим самим кольором. Це полегшує редагування і перевірку правильності формул.

Отже, крім чисел у формулах можна використовувати посилання – адреси інших комірок. Наприклад, =A4/A5+10 чи =B8+E9. В формулах електронної таблиці використовуються два типи адрес:

Абсолютна адреса – не змінюється при копіюванні формул, наприклад, \$F\$1, знак \$ фіксує букву стовпчика і цифру рядка.

Відносна адреса – адреса, яка при копіюванні формул коректується.

В основному посилання на клітинки є відносними — при копіюванні формули адреси в посиланнях автоматично змінюються у відповідності з відносним розташуванням вихідної клітинки і створеної копії. Проте трапляються випадки, коли адреси деяких клітинок не потрібно змінювати при копіюванні формул. Тоді використовуються абсолютні посилання.

Для автоматизації обчислень при роботі з електронними таблицями використовують автозаповнення формулами. Ця операція виконується наступним чином. У правому нижньому куті рамки активної клітинки є чорний квадратик (маркер автозаповнення). При наведенні на нього вказівник миші набуває форми чорного хрестика. Перетягування маркера заповнення розглядається як операція «розмноження» вмісту клітинки в гори-



зонтальному чи вертикальному напрямі. При автозаповненні формул до уваги приймається характер посилань у формулах.

При виконанні будь-якої роботи помилки – явище поширене. (☺) Тому необхідно знати, як реагує Microsoft Excel на наявність помилок.

Повідомлення про помилки:

Якщо формула у комірці не може бути правильно обчислена, Microsoft Excel виводить у комірку повідомлення про помилку. Якщо формула містить посилання на комірку, що містить значення помилки, то замість цієї формули також буде виводитися повідомлення про помилку.

Значення помилок:

– ширина комірки не дозволяє відобразити число в заданому форматі;

#ИМЯ? – Microsoft Excel не може розпізнати ім'я, використане в формулі;

#ДЕЛ/0! – у формулі робиться спроба поділу на нуль;

#ЧИСЛО! – порушені правила задавання операторів, прийняті в математиці;


#Н/Д – таке повідомлення може з'явитися, якщо в якості аргументу задане посилання на порожню комірку;

#ПУСТО! – невірно зазначене перетинання двох областей, що не мають загальних комірок;

#ССЫЛКА! – у формулі задане посилання на неіснуючу комірку;

#ЗНАЧ! – використаний неприпустимий тип аргументу.

Швидкий спосіб обчислення суми по стовпчиках і стрічках.

Кнопка "Автосумма", яка розташована на панелі інструментів Стандартная, використовується для швидкого підрахунку сум даних: .

Засіб "Автосумма" можна використовувати в таких випадках:

- Для складання значень одного стовпчика чи стрічки;
- Для складання значень декількох стовпчиків і стрічок;
- Для складання значень декількох стовпчиків і декількох стрічок і обчислення загальної суми.

4. Практичне завдання. Інструктаж з техніки безпеки.

1. Відкрити книгу "Товари".

2. В таблиці "Відомість обігу товарів" всі комірки, виділені сірим кольором, необхідно заповнити даними. Але всі дані повинні бути знайдені в результаті застосування формул. При виконанні подібних обчислень в межах одного стовпчика (окрім "усього") зручно користуватись автозаповненням. А в рядку з "усього" – зручно рахувати з допомогою функції "Автосума".

3. На аркуші 2 створити ще одну таблицю "Рахунок-фактура" за зразком. Так само всі комірки, виділені сірим кольором, необхідно заповнити даними. Виділяти сірим кольором необов'язково.



4. Прозвітувати про виконання завдання вчителю.

5. Зберегти результати роботи і закрити всі вікна.

Рахунок-фактура

Код товару	Найменування товару	К-ть товару, кг	Ціна, грн	Сума, грн	Торговельна знижка, %	Сума з урахуванням знижки, грн
156	<i>М'ясо</i>	20	20		6	
36	<i>Риба</i>	23	16		3	
956	<i>Сир</i>	65	30		4	
753	<i>Цукор</i>	95	3		2	
Всього						

6. Підведення підсумків роботи.

При необхідності обговорюються проблеми та помилки, що виникали під час роботи

7. Закріплення вивченого матеріалу.

- З яких компонентів може складатися запис формул?
- Правила введення формул до ЕТ.
- Які оператори Excel вам відомі і порядок їх виконання?

8. Домашнє завдання: Опрацювати конспект уроку.

Р.1, §4, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов.

§ 12.7 "Курс інформатики". В.Д. Руденко, О.М. Макарчук, М.О. Патланжоглу.

9. Підсумок уроку.



Урок № 11

<i>Тема:</i>	Використання функцій в ЕТ.
<i>Мета:</i>	Ознайомити з правилами написання та порядком введення функцій, з логічними та математичними функціями в ЕТ Ехсел. Сформувати в учнів практичні навички використання функцій в ЕТ Ехсел. Вдосконалити навички роботи з аркушами.
<i>Базові поняття й терміни:</i>	Функція, вставка функції,

Хід уроку

1. Організаційний етап.
3. Оголошення теми і мети уроку.
3. Вивчення нового матеріалу.

Розповідь вчителя з використанням демонстрації по локальній мережі.

Крім чисел, текстів і адрес комірок у формулу можуть входити вбудовані в Ехсел функції.

Функція - це вже готова складна формула, за допомогою якої виконуються розрахунки над групою даних визначеного типу.



Кожна функція має ім'я, аргументи, поміщені в круглі дужки, і значення, що обраховується. Функцію можна розуміти як заздалегідь визначену послідовність дій, що виконується над аргументами. Результатом виконання цих дій є значення, що обраховується. Аргументи відокремлюються один від одного крапкою з комою. Наприклад, функція СУММ(А1;В1;100) обчислює суму трьох значень: вмісту комірок А1, В1 і числа 100.

Для введення функції у комірку необхідно:

- виділити комірку для формули;
- викликати *Мастер функций* за допомогою команди *Функция* з меню

Вставка

- у діалоговому вікні, що з'явилося, вибрати тип функції в переліку *Категория* і потім потрібну функцію в переліку *Функция*;
- натиснути кнопку ОК;
- у полях *Число1*, *Число2* та ін., наступного вікна ввести аргументи функції (числові значення або посилання на комірки);

- щоб не вказувати аргументи з клавіатури, можна натиснути кнопку , яка знаходиться праворуч поля, і виділити мишею ділянку комірок, що містять аргументи функції; для виходу з цього режиму слід натиснути кнопку , яка знаходиться під рядком формул;

- натиснути ОК.



В Excel пропонується для вибору такі категорії функцій:

- Статистичні;
- Повний алфавітний перелік;
- Посилання й масиви;
- Фінансові;
- Робота с базою даних;
- Дата й час;
- Текстові;
- Математичні;
- Логічні;
- Перевірка властивостей й значень.

Назва категорії відповідає змісту, властивостям функцій. Математичні функції дозволяють проводити прості і складні обчислення, наприклад обчислення суми діапазону комірок, обчислення суми комірок діапазону, які задовольняють вказаній умові, округлення чисел та ін.

Також часто використовуються і вивчаються логічні функції, тому що в деяких задачах значення клітинки (чи виразу) залежить від деякої умови.

Логічні умови використовують знаки $>$, $<$, $=$, $>=$, $<=$, $<>$.

ЕСЛИ (логічна умова; дія при умові ІСТИНА; дія при умові ХИБНА).

И(умова1;умова 2;умова3;...)

ИЛИ(умова1;умова2;...)

Статистичні функції дають змогу обрахувати деякі терміни, відомі вам з прикладної математики: моду, медіану тощо.

До речі, в комірку можна вставити текстову або навіть звукову примітку – певний опис або зауваження щодо її вмісту. Для цього використовується команда меню *Вставка - Примечание*.

4. Усвідомлення і закріплення учнями вивченого матеріалу

Учням пояснюється і демонструється застосування функцій на прикладах.

1) Знаходження значення виразу $\sqrt{\frac{1}{x} + \sin(x)} - x^6$. Використовуються функції *корінь*, синус (*sin*) та *ступінь*.

2) Знаходження коренів квадратного рівняння з використанням логічної функції *Если*.

	A	B	C	D	E	F
1						
2	a	b	c		d	
3	1	4	-5		36	
4						
5		x1	x2			
6		1	-5			
7						

Комірка B6 містить формулу:

=Если(E3>=0;(-B3-Корень(E3))/2*A3;"розв'язку немає")



а комірка С6: = Если(Е3>=0;(-В3+Корень(Е3))/2*А3;" ").

Як видно, якщо дискримінант менший за нуль, в комірці В6 з'явиться повідомлення "розв'язку немає", а в С6 буде порожня клітинка:

	A	B	C	D	E	F
1						
2	a	b	c		d	
3	1	4	9		-20	
4						
5		x1	x2			
6			розв'язку немає			
7						
8						

Увагу учнів слід звернути на те, що така задача може бути розв'язана багатьма способами.

5. Виконання практичного завдання.

Інструктаж з техніки безпеки.

1. Відкрити книгу "Товари".
2. На аркуші 3 створити таблицю "Оплата праці" згідно зразка.
3. Комірки, виділені сірим кольором, необхідно заповнити даними за такою схемою:

- стаж до 10 років – доплати немає
- стаж >10, але <20 – доплата 10%
- стаж >20 – доплата 20%

До видачі – це сума з урахуванням доплат.

Оплата праці

Прізвище, ім'я та по батькові	Тарифна ставка	Відроблено днів	Стаж	Доплата за стаж в %	До видачі
Іванов І.І.	9	26	16		
Сидоров С. С.	7	24	29		
Петров П. П.	11	28	7		

6. Закріплення матеріалу, вивченого на уроці.

- Для чого в електронних таблицях використовуються вбудовані функції?
- Перелічіть функції, які ви використовували під час цього уроку?
- Які логічні функції, вбудовані в Excel ви знаєте?
- Наведіть приклади використання логічних функцій.
- Що називається логічним виразом?
- Як скопіювати функцію в інші комірки ЕТ?
- Скільки логічних функцій можуть бути вкладеними одна в одну?

7. **Домашнє завдання:** Опрацювати конспект уроку. § 12.7 "Курс інформатики". В.Д. Руденко, О.М. Макарчук, М.О. Патланжоглу.

Повт. Р.1, §4, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов.

8. Підсумок уроку.

**Урок № 12**

Тема:	Діаграми в електронних таблицях.
Мета:	Ознайомити з правилами побудови діаграм і графіків на основі табличних даних, з використанням майстра створення діаграм в ЕТ Excel. Навчити учнів будувати діаграми та графіки в електронних таблицях. Вдосконалити навички роботи з аркушами.
Базові поняття й терміни:	Діаграма, графік, види діаграм.

Хід уроку**1. Організаційний етап.****2. Відтворення і корекція опорних знань учнів.**

- З яких компонентів може складатися запис формул?
- Які оператори Excel вам відомі?
- Як ввести функцію до формули за допомогою Майстра функцій?
- Чи можна ввести формулу вручну?
- Для чого в електронних таблицях використовуються вбудовані функції?
- Перелічіть функції, які ви використовували під час минулого уроку.
- Які логічні функції, вбудовані в Excel ви знаєте?
- Що називається логічним виразом?
- Наведіть приклади використання логічних функцій.

3. Мотивація навчальної діяльності учнів.

В різних ситуаціях фахівця можуть цікавити як самі числа, показані в таблиці, так і тенденції у їх зміні, які показують діаграми. Отже, діаграми призначені для графічного відображення числових даних.

За допомогою діаграм, на яких зображені зміни у виробництві, аналізують економічний стан на підприємстві. Аналіз такої інформації дає можливість прийняти правильне рішення і підвищити конкурентну здатність продукції, яка виробляється, зробити її дешевшою і якіснішою.

В ЕТ Excel вмонтовані спеціальні засоби, що носять назву ділової графіки, які дозволяють зобразити табличні дані у графічному вигляді.

4. Оголошення теми і мети уроку.**5. Вивчення нового матеріалу.****а) Лекція з елементами демонстрації.**

Діаграми служать для графічного представлення даних на листі. В Excel існує ряд стандартних типів діаграм. Діаграми можна розміщувати прямо в листі поряд з даними, котрі використовувались для побудови діаграм. Крім того, діаграма може займати в книзі окремий лист, який називається діаграмним листом.



Незалежно від того, яким чином була створена діаграма, вона завжди зв'язана з даними листа. При зміні даних діаграма буде автоматично оновлюватися.

Діаграми можна створювати двома способами – за допомогою швидких клавіш і за допомогою майстра діаграм. Перший метод виконується швидше і створює стандартну двовимірну гістограму. Другий спосіб дозволяє модифікувати діаграму в процесі її створення.

☞ **Діаграма зберігає зв'язок з даними на листі. Якщо дані змінюються, автоматично відбувається оновлення діаграми.**

Кожна діаграма в Excel складається з ряду об'єктів.

Об'єкти діаграми в Excel.

№	Об'єкт діаграми	Опис
1	<i>Ось категорій</i>	Вісь, на якій відображаються заголовки з листа.
2	<i>Метки даних</i>	Додаються в діаграму для відображення конкретного значення точки даних.
3	<i>Точка даних</i>	Елемент ряду даних, який відповідає значенню однієї комірки в листі.
4	<i>Ряд даних</i>	Рядок чи стовпчик даних з листа. Назви всіх рядів приводяться в легенді.
5	<i>Таблиця даних</i>	Таблиця, яка додається нижче вісі категорій, в якій відображаються вхідні дані діаграми.
6	<i>Линии сетки</i>	Лінії, які починаються з ділень шкали. Як правило, їх додають до вісі значень, щоб ідентифікувати значення точок даних.
7	<i>Область построения диаграммы</i>	Прямокутна область, яка обмежена осями.
8	<i>Легенда</i>	Текстове поле з описом рядів даних.
9	<i>Шкала</i>	Цифрові поділки на осі значень. Мінімальне значення=0.
10	<i>Маркеры выделения</i>	При клацанні на об'єкті біля нього з'являється декілька чорних квадратів, які показують, що об'єкт вибраний. Об'єкт вибирають для переміщення чи редагування.
11	<i>Метки делений</i>	Маленькі лінії, які відображають ділення шкали по осях категорій і значень.
12	<i>Заголовки</i>	Назва діаграми, назва осі значень, назва осі категорій
13	<i>Ось значений</i>	Вісь, на якій розташовуються значення даних з листа. Вона звичайно вертикальна.

В Excel крім 14-ти стандартних типів діаграм, існує іще 20 нестандартних.

Стандартні типи діаграм, представлені в Excel.

Тип діаграми	Розмірність	Описання
<i>Гистограма</i>	Двовимірна та тривимірна	Дані відображаються у вигляді вертикальних смуг. Найбільш поширений тип. В Excel використовується за замовчуванням. Представлена у наступному вигляді: об'ємна, з накопиченням і нормована на 100 %.



<i>Линейчатая</i>	Двовимір-на та тривимір-на	Дані відображаються у вигляді горизонтальних смуг. Корисна під час порівняння величин за один часовий проміжок та у випадках, коли підписи категорій надто довгі. Представлена у наступному вигляді: об'ємна, з накопиченням і нормована на 100 %.
<i>Коническая, цилиндрическая, пирамидальная</i>	Тривимір-на	Привабливі тривимірні варіанти гістограми та лінійчатої діаграми.
<i>График</i>	Двовимір-на та тривимір-на	Дані відображаються у вигляді точок, що з'єднані між собою лініями. Цей тип діаграм використовується для відображення зміни даних у часі. Представлена у наступному вигляді: об'ємна і з накопиченням.
<i>С областями</i>	Двовимір-на та тривимір-на	Подібна до графіку, тільки область під лінією зафарбована. Використовується для відображення великої кількості точок. Має наступні різновиди: об'ємна і з накопиченням.
<i>Круговая</i>	Двовимір-на та тривимір-на	Відображає один ряд чи категорію даних. Використовується для того, щоб показати, скільки відсотків складає кожна точка даних від загального додатку. Цей тип діаграм представлений наступними видами: об'ємна, розрізна, вторинна кругова. Вторинні діаграми використовуються у тих випадках, коли частину елементів необхідно відокремити від загальної суми і представити їх в окремій коловій діаграмі або гістограмі.
<i>Кольцевая</i>	Двовимір-на	Схожа на колову. Використовується для відображення відсотка від загальної суми, але для кількох рядів. Дані відображаються у вигляді кілець. Можливе створення розрізної кільцевої діаграми.
<i>Точечная</i>	Двовимір-на	Відображає маркери для кожної точки даних. Застосовується для відображення розподілу даних, що часто використовується у наукових роботах. Представлена наступними видами: тільки точки; точки, поєднані згладжуючими лініями; точки, поєднані відрізками.
<i>Пузырьковая</i>	Двовимір-на	Нагадує точкову діаграму, у якій для кожної точки відображається мітка. Чим більше значення, тим більша бульбашка. Може бути представлена в об'ємному вигляді.
<i>Биржевая</i>	Двовимір-на	Використовується для відображення зміни курсу біржових цін. Представлена наступними різновидами: для вибору з трьох значень (самий високий курс, самий низький курс і курс закриття), для вибору з чотирьох значень (курс відкриття, самий високий курс, самий низький курс, курс закриття), ще для одного набору з чотирьох значень (об'єм, самий високий курс, самий низький курс і курс закриття) та для набору для вибору з п'яти значень (об'єм, курс відкриття, самий високий курс, самий низький курс і курс закриття).
<i>Поверхность</i>	Тривимір-на	Подібно до топографічних картах, ця діаграма відображає "підвищення" та "западини" для сукупності даних. Різновид цього типу діаграм – контур.



Лепестко- вая	Двовимір- на	Вісь значень представлена проміннями, що виходять із загального центру, а точки даних поєднані відрізками, утворюючи структуру, що нагадує павутиння. Можливий варіант діаграми з полями.
------------------	-----------------	---

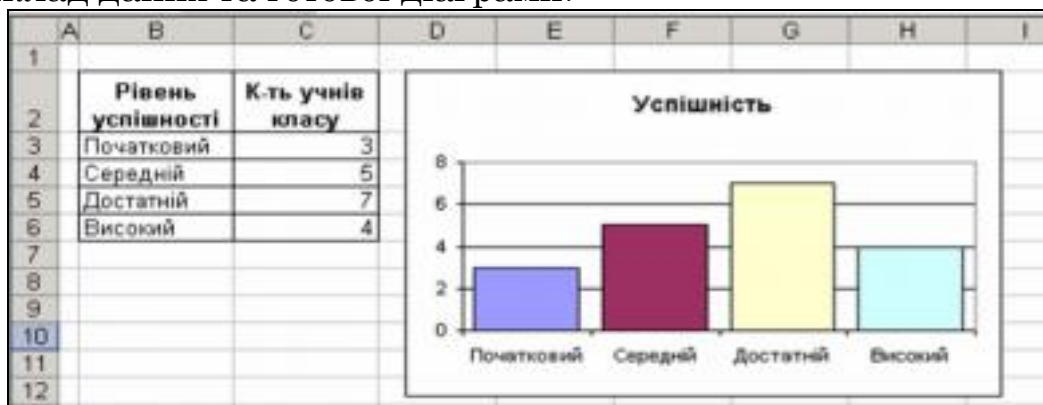
Звичайно ж, з часом приходиться формувати готові діаграми.

Параметри форматування елементів діаграм Excel.

Параметр фо- рмування	Опис
<i>Вид</i>	Набір опцій змінюється від елемента до елемента. Включає в себе колір меж чи осі, тінь, колір заливки і способи заливки, ділення.
<i>Ось</i>	Додавання додаткової вісі. Використовується, в основному, для комбінованих діаграм.
<i>У-погрешность</i>	Відображення похибок. Можна вибрати величину похибки, включаючи фіксоване значення, відносне значення, стандартне відхилення, стандартну похибку чи своє значення.
<i>Подписи данных</i>	Відображення підписів даних, значень чи процентів від загальної суми.
<i>Порядок рядов</i>	Зміна порядку рядів.
<i>Параметры</i>	Різні опції, що змінюються від одного елемента до другого.
<i>Шкала</i>	Використовується для вісі значень. Можна вибрати мінімальне значення, максимальне значення, ціну основного ділення.
<i>Шрифт</i>	Тип, накреслення, розмір, підкреслення, колір і фон.
<i>Число</i>	Грошовий, процентний, дробовий і додатковий формати чисел.
<i>Выравнивание</i>	Дозволяє обертати текст від -90° до 90° чи виводити текст вертикально.
<i>Размещение</i>	Використовується для легенди. Дозволяє вибрати місце на діаграмі, де буде показана легенда.

Оскільки матеріалу теоретичного досить багато, з вищенаведеними таблицями учнів потрібно просто ознайомити, а обов'язково відділити час для демонстрації створення діаграми, користуючись *Мастером діаграм*.

Приклад даних та готової діаграми:



5. Осмислення набутих знань.

а) Гра "Закінчи речення". Найповніші відповіді оцінюються.

- Діаграма ET – це ...

... графічний спосіб подання табличних даних. Дані робочого аркуша, що використовуються при створенні діаграми, зв'язуються з нею, і при їхній зміні діаграма оновлюється.



- Для побудови діаграм можна використати дані, ...
... які знаходяться в несуміжних комірках або діапазонах.
- Можливі такі види розміщення діаграм: ...
... на окремому аркуші, на поточному аркуші.

Список питань можна продовжувати в залежності від наявності вільного часу.

б) Практичне завдання. Інструктаж з техніки безпеки.

- Запустити табличний процесор Excel.

Завдання 1.

- Створити кругову діаграму розподілу акцій компанії "Vektor" між власниками за наведеними нижче даними.

№	Власник акцій	Кількість акцій
1	Петренко А.А.	234
2	Василюк В.Н.	554
3	Корейко М.М.	78
4	Коваль Б.Ф.	789
5	Бондаренко В.К.	342
6	Артеменюк І.С.	54
7	Шевченко Л.С.	178

- Діаграма повинна включати легенду, поміщену справа, де кожен колір буде відповідати прізвищу власника.

Завдання 2.

- Створити графік коливання курсу валюти за рік за наведеними нижче даними.

	Місяці											
	1	2	3	4	5	6	7	8	9	10	11	12
Значення	3,65	3,84	3,73	4,00	4,20	4,25	5,15	5,00	5,25	5,00	6,00	6,24

7. Закріплення вивченого на уроці матеріалу.

Експрес - опитування

- Які способи побудови діаграми ви знаєте?
- Які види розміщення діаграм ви знаєте?
- Які види діаграм ви знаєте?
- Як називається діаграма з якою ви працювали на цьому уроці?
- З яких етапів складається побудова діаграм та графіків в ЕТ?
- Що називають легендою діаграми?
- Як зробити надписи з даними на діаграмі?

8. **Домашнє завдання:** Опрацювати конспект уроку. § 12.8 "Курс інформатики". В.Д. Руденко, О.М. Макарчук, М.О. Патланжоглу.

9. Підсумок уроку.



Урок № 13

Тема: Упорядкування, Фільтрація та пошук даних.

Мета: Ознайомити з правилами впорядкування і пошуку даних в середовищі табличного процесора, з різними способами консолідації даних ЕТ Excel.

Навчити учнів створювати і застосовувати фільтри різних типів.

Базові поняття й терміни: Упорядкування, фільтрація, пошук, консолідація, автофільтр.

Хід уроку

1. Організаційний етап.

2. Відтворення і корекція опорних знань учнів.

Інтелектуальна розминка.

- Які способи побудови діаграми ви знаєте?
- Які види розміщення діаграм ви знаєте?
- Які види діаграм ви знаєте?
- Як називається діаграма з якою ви працювали на цьому уроці?
- Які основні об'єкти діаграми?
- Які основні типи діаграм.?
- Як працювати з *Мастером діаграмм*?

3. Мотивація навчальної діяльності учнів.



Часто виникає потреба впорядкувати дані: чи відобразити прізвища в алфавітному порядку, чи поставити числа в порядку зростання. Як правильно це зробити – ми вивчатимемо на сьогоднішньому уроці.

4. Оголошення теми і мети уроку.

5. Вивчення нового матеріалу.

Розповідь вчителя з використанням демонстрації по локальній мережі.

Сортування даних.

Якщо виникає потреба якось впорядкувати дані, відобразити в алфавітному порядку, поставити числа в порядку зростання, то для цього на панелі інструментів є кнопки  або 

Також можна скористатися командою меню *Данные-Сортировка*.

Діалогове вікно *Сортировка диапазона* служить для вибору поля, на якому виробляється сортування. Якщо задані заголовки полів, то як позначення використовуються саме вони, якщо заголовки відсутні, як заголовки використовуються позначення стовпців. Перше з обраних полів визначає режим первинного сортування. Якщо при збігу значень у цьому



полі існують додаткові критерії, що визначають порядок записів, то можна також задати поля для *вторинного* і *третинного* сортування. На кожному з полів сортування може виконуватися в порядку зростання чи зменшення.

Фільтрація бази даних. Тому що база даних може включати величезне число записів (у програмі Excel природною межею служить максимальне число рядків робочого аркуша — 65536), не завжди потрібно відображати всі ці записи. Виділення підмножини загального набору записів називається *фільтрацією*.

Найбільш простим способом фільтрації в програмі Excel є використання *автофільтра*.

Застосування автофільтра. Включення режиму фільтрації здійснюється командою *Данные - Фильтр - Автофильтр*. При цьому для кожного поля бази даних автоматично створюється набір стандартних фільтрів, доступних через списки, що розкриваються. Кнопки цих списків, що розкриваються, відображаються біля поля заголовка кожного стовпця.

За замовчуванням використовується варіант *Все*, який вказує, що записи бази даних повинні відображатися без фільтрації. Варіант *Первые 10* дозволяє відібрати визначене число (чи відсоток) записів за яким-небудь критерієм. Варіант *Условие* дозволяє задати спеціальну умову фільтрації. Крім того, є можливість добору записів, що мають у потрібному полі конкретне значення.

При виборі розширеної фільтрації командою *Данные - Фильтр - Расширенный фильтр* можна виконати фільтрацію на місці чи витягти відфільтровані записи і помістити їх окремо, на будь-який робочий аркуш будь-якої відкритої робочої книги.

Консолідація даних.

- Консолідація даних з використанням тривимірних посилань;
- Консолідація даних за розміщенням;
- Використання майстра шаблонів;
- Консолідація даних по категоріям;
- Консолідація даних шляхом створення звіту зведеної таблиці.

Досить корисним в обробці таблиць є знаходження *Итогов* в таблицях. Підсумки використовують для знаходження різних показників (суми, середнього значення, мінімального та максимального значень та ін.).

Для отримання підсумків до таблиці застосовують команду *Данные - Итоги*. У вікні, що з'явилося, задають:

- назву поля з об'єктами, для яких створюють підсумки.
- потрібну операцію.
- назву поля, що містить дані для підсумків.



6. Практичне завдання. Інструктаж з техніки безпеки.

- Відкрити книгу "Товари".
- Створити подану нижче таблицю, помістити її на четвертий аркуш книги.

Назва	Ціна
Вуглекислота	35,00
Шпатлівка	6,00
Труба 10*20*1,5	0,92
Труба 18*1,5	0,47
Труба 25*40	0,75
Труба 12*1,5	0,30
Труба 25*28*1,5	1,25
Труба 60*30*3	4,03
Емаль блакитна	16,50
Емаль жовта	5,00
Емаль біла ПФ	3,53
Коло відрізне	3,56
Проволока 4мм	0,96
Шпатлівка	6.00

- Впорядкувати дані в першому рядку за алфавітом.
- Продемонструвати результат учителеві.
- Виконати пошук даних (фільтрацію) у створеній таблиці. Вибрати в меню *Данные* пункт *Фильтр*, звідки підпункт *Автофильтр*.
- Вказати умову, за допомогою якої на екрані буде відображатись тільки товар із назвою "Вуглекислота".
- Продемонструвати результат учителеві.
- Відмінити ці дії, для чого вибрати в цьому ж спискові пункт (*Все*).

7. Закріплення вивченого.

- Навіщо потрібен фільтр?
- Які є види фільтрів в ЕТ Excel?
- Як ви розумієте поняття "консолідація даних"?

8. Домашнє завдання: Опрацювати конспект уроку.

9. Підсумок уроку.

**Урок № 14**

Тема: Виведення табличних даних та діаграм на друк. Узагальнення і систематизація знань.

Мета: Сформувати вміння з допомогою принтера роздруковувати готові матеріали.

Підготувати учнів до тематичного оцінювання.

Базові поняття й терміни: Друкування книг та аркушів ЕТ, орієнтація аркуша, колонтитули.

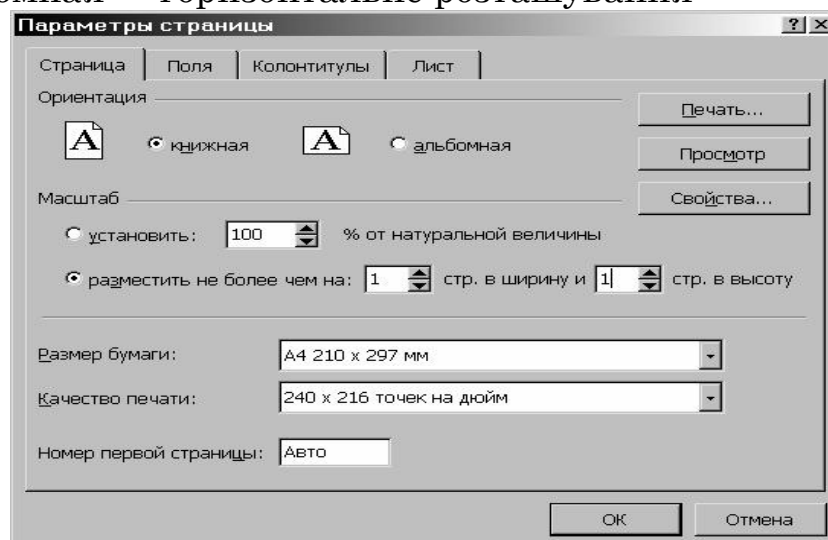
Хід уроку**1. Організаційний етап.****2. Мотивація навчальної діяльності учнів.**

Екранне представлення електронної таблиці в Excel значно відрізняється від того, що виходить при виведенні даних на друк. Це зв'язано з тим, що єдиний робочий аркуш приходиться розбивати на фрагменти, розміри яких залежать від формату друкованої сторінки. Крім того, елементи оформлення робочого вікна програми: номери рядків і стовпців, умовні границі клітинок — звичайно не відображаються при друці.

3. Оголошення теми і мети уроку.**4. Вивчення нового матеріалу.****Розповідь вчителя.****Друкування таблиць**

Перед друкуванням таблиць необхідно встановити параметри сторінки за допомогою команди *Параметри сторінки* меню *Файл*. На вкладці *Страница* встановлюється орієнтація і розміри аркушу паперу, масштаб зображення і якість друкування. У пункті *Ориентация* встановлюється орієнтація аркушу:

- "Книжная" – вертикальне розташування аркушу паперу;
- "Альбомная" – горизонтальне розташування





Якщо встановлений прапорець "Установить", то таблиця буде зображена при друкуванні в масштабі, зазначеному в полі праворуч (без зміни розміру аркушу на екрані).


Якщо встановлений прапорець "Разместить не более чем на", то таблиця або виділення буде зменшено так, щоб результат заповнював у точності визначену кількість сторінок.

У полі "Размер бумаги" вибирається формат аркушу паперу, а в полі "Качество печати" – якість друку.

Для установаження полів сторінки слід вибрати вкладку "Поля" та вказати потрібні значення полів. У полях "Верхнего колонтитула" і "Нижнего колонтитула" вибирається відстань від верхнього краю аркушу до верхнього колонтитулу і нижнього краю аркушу до нижнього колонтитулу відповідно.

Для введення колонтитулів використовується вкладка "Колонтитулы". Вміст колонтитулів вводиться після натискання кнопок "Создать верхний колонтитул" і "Создать нижний колонтитул".

Після встановлення всіх параметрів у вікні "Параметры страницы" необхідно натиснути кнопку ОК.

Для перегляду таблиці перед друком слід вибрати пункт "Предварительный просмотр" у меню "Файл" або натиснути кнопку . У цьому режимі, щоб збільшити зображення слід навести покажчик миші, який матиме вигляд лупи, на потрібний фрагмент і натиснути кнопку миші. Наступне натискання кнопки миші віддаляє зображення. Вийти з режиму попереднього перегляду можна за допомогою кнопки "Закреть" або клавіші "Esc".

Для друкування таблиці необхідно в меню "Файл" вибрати пункт "Печать".

5. Практичне завдання. Інструктаж з техніки безпеки.

Кожен учень повинен віддрукувати один лист з книги "Товари" а також той, на якому знаходиться діаграма.

При цьому всі опції друку встановлюються самостійно.

☞ Мережний або локальний принтер повинен бути налаштований вчителем заздалегідь.

6. Закріплення вивченого.

- Як роздрукувати книгу в програмі Microsoft Excel?
- Як встановити верхні та нижні колонтитули?
- Як провести розмітку сторінки для друку?

7. Домашнє завдання: Опрацювати конспект уроку.

Р.1, §5, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов, § 12.4 "Курс інформатики". В.Д. Руденко, О.М. Макарчук, М.О. Патланжоглу.

8. Підсумок уроку.



Урок № 15

<i>Тема:</i>	Тематичне оцінювання з теми "Табличний процесор"
<i>Мета:</i>	Перевірка та контроль знань, вмінь і навичок учнів.

Хід уроку

1. Організаційний етап

2. Оголошення теми і мети уроку.

3. Тематичне оцінювання.

1. *Комп'ютерне тестування.* (Використовується програма Test-W або аналогічна). *Інструктаж з техніки безпеки.*

2. Практичне завдання.

- Створіть таблицю з 3-х стовпчиків: прізвище, ім'я, по батькові, заповніть її даними, створивши 6 записів.
- Вставте на початку таблиці новий стовпчик та зробіть нумерацію.
- Додайте в кінці даної таблиці стовпчик "Місячна заробітна плата", заповніть його даними (100, 200, 300,).
- В комірці під останньою зарплатою обчисліть за допомогою майстра функцій максимальну зарплату.
- Додайте стовпчик "ПДВ", за допомогою логічних функцій створіть формулу для нарахування ПДВ (він становить 15%, якщо зарплата менше 250 грн, і 20%, якщо більша чи рівна 250 грн.).
- Додайте стовпчик "Нараховано", в якому обчисліть зарплату без ПДВ.
- Під останньою коміркою цього стовпчика підрахуйте суму нарахованої зарплати.
- На новому листі створіть двохстрічну таблицю з такими даними: день тижня та кількість уроків в цей день. Побудуйте гістограму для цієї таблиці, виведіть на діаграму її назву.
- Збережіть книгу під назвою "ТО Прізвище" (Прізвище пише кожен своє).

4. Домашнє завдання: Опрацювати конспект уроку.

Повт Р.1, "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов, § 12 "Курс інформатики". В.Д. Руденко, О.М. Макаруч, М.О. Патланжоглу.

5. Підсумок уроку.



Урок № 16

<i>Тема:</i>	Бази даних. Системи управління базами даних. Основні об'єкти баз даних.
<i>Мета:</i>	Розглянути види баз даних та необхідність їх використання, структуру MS-Access. Ознайомити учнів з методикою проектування баз даних.
<i>Базові поняття й терміни:</i>	База даних, СУБД, MS-Access, види баз даних, функції БД, об'єкти бази даних, проектування бази.

Хід уроку

1. Організаційний момент.

Тема, яку ми починаємо вивчати має назву "Бази даних. Системи управління базами даних". Складається вона з 8 годин (уроків), по закінченню – тематичне оцінювання у вигляді практичної роботи. Ця тема є останньою в розділі "ППЗ загального призначення".

На уроках ми будемо вивчати теоретичний матеріал, виконувати різноманітні практичні завдання. В результаті ви повинні отримати певні знання, вміння та навички. (Стенди на зразок "Вивчаємо тему", посібники "Учні повинні знати та вміти", "Критерії оцінювання").

2. Мотивація навчальної діяльності учнів.

Під час інформаційної діяльності людина збирає і накопичує відомості про довкілля. До появи обчислювальної техніки вся інформація зберігалася в письмовому або надрукованому вигляді. Однак зі збільшенням обсягів інформації загострювалося питання зберігання інформації та її обробки. Нагадаємо, що інформація для обробки називається *даними*.

Щоб користувач легко міг знаходити потрібну інформацію, вона має бути організована певним чином. Це стосується не лише інформації у комп'ютері, а й будь-якої інформації про об'єкти реального світу. Скажімо, зручно знаходити потрібну книгу в бібліотеці, користуючись каталогом. Легко відшукати у газеті оголошення, що вас цікавлять. Така легкість пошуку можлива завдяки тому, що дані у каталозі або у газеті мають структуру, або, інакше, *структуровані*. Всі книги описані однаково: автор, назва, видавництво, рік видання тощо. Всі оголошення з продажу розміщені по рубриках і також мають визначену структуру: короткий опис товару, ціна, телефон.

Під час вивчення цієї теми ми якраз і будемо вивчати, як інформацію об'єднати згідно певних критеріїв, та як працювати з такою інформацією.

3. Оголошення теми уроку.



4. Вивчення нового матеріалу.

Розповідь вчителя з використанням демонстрації по локальній мережі.

База даних (БД) – це систематизоване сховище інформації з певної предметної області, до якого можуть мати доступ багато прикладних програм.

Але якщо зберігання і обробка даних відбувається з допомогою комп'ютера, то для цього нам потрібні певні програмні засоби - системи управління базами даних.

Система управління базами даних (СУБД) – це програмні засоби для створення, введення і використання БД. Всі існуючі системи задовольняють, як правило, таким вимогам:

- Можливість маніпулювання даними
- Можливість пошуку і формування запитів.
- Забезпечення цілісності (узгодженості) даних.
- Забезпечення захисту і таємності.

Приклади СУБД: dBASE, FoxBase, FoxPro, MS Access, Paradox та ін.

На наших уроках ми вивчатимемо теми на прикладі СУБД Access, яка входить до складу пакету Microsoft Office. За структурою організації інформації в БД розрізняють такі моделі баз даних:

- реляційна;
- ієрархічна;
- мережева.

База даних СУБД Access є реляційною базою даних, яка складається з взаємопов'язаних двовимірних таблиць.

СУБД Access дає змогу:

- Проектувати табличні об'єкти бази даних;
- Встановлювати зв'язки між таблицями;
- Вводити, зберігати, переглядати, сортувати, модифікувати дані;
- Створювати й використовувати об'єкти БД.

Об'єкти СУБД Access:

Бази даних – файл, який містить різні об'єкти збереження даних.

Таблиця – організація збереження даних у вигляді двовимірного масиву. Вона є основним об'єктом БД. Решта – похідні від таблиці.

Форми – об'єкти для зображення даних із таблиці на екрані в зручному для перегляду та обробки вигляді.

Запити – об'єкти для вибору та фільтрації даних таблиці за визначеними критеріями.

Звіт – формування документа.

Макроси – опис дій у вигляді послідовності команд та їх автоматичного виконання.

Модулі – програми на Visual Basic, які розробляє користувач для реалізації нестандартних процедур.



Але це все стосується готових баз даних. А ми сьогодні повинні також вивчити, з чого починається створення баз.

Створення бази даних відбувається зазвичай в кілька етапів:

➤ Проектування (на папері або в спеціальних програмах) – є самим важливим етапом, в ході якого потрібно:

- визначити проблему;
- вибрати та проаналізувати інформацію, яку отримуватиме конкретна база даних;
- подати інформацію в базу даних.

➤ Програмна реалізація – технологія створення бази даних;

➤ Експлуатація БД.

☞ Перед роботою в Access таблиці (незаповнені) краще зобразити на папері, це зменшить кількість виправлень у готовій таблиці.

5. Закріплення вивченого теоретичного матеріалу.

Учням пропонується дата відповіді на запитання, при цьому дозволяється попарне обговорення (в межах парти):

- Як взаємопов'язані поняття БД та СУБД?
- Яка з моделей баз даних є провідною?
- Які основні функції СУБД?
- З яких об'єктів складається СУБД Access?

Наприкінці учні визначають основні функції СУБД:

Визначення даних – можна визначити, яка саме інформація буде зберігатися в базі даних, задати структуру даних та їх тип (наприклад, максимальну кількість цифр або символів), а також вказати, як ці дані зв'язані між собою. В деяких випадках ви можете також задати формати та критерії перевірки даних;

Обробка даних – дані можна обробляти найрізноманітнішими способами. Можна об'єднувати дані з іншою зв'язаною з ними інформацією та обраховувати стогові значення;

Управління даними – можна вказувати, кому дозволено знайомитися з даними, корегувати їх або додавати нову інформацію. Можна також визначити правила колективного користування даними.

6. Домашнє завдання. Опрацювати конспект уроку.

Р.2, §1-3 вивчити, §4 – прочитати Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов,

§ 13 "Курс інформатики". В.Д. Руденко, О.М.Макарчук, М.О. Патланжоглу.

7. Підведення підсумків уроку.

Розглянувши та вивчивши всі теми та виконавши практичні завдання, ви зможете вільно орієнтуватися в такій системі управління базами даних як Access, та використовувати набуті навички для продовження вдосконалення своїх знань цієї теми.

**Урок № 17**

Тема:	Створення баз даних. Таблиці.
Мета:	Закріпити знання про структуру баз даних у MS-Access". Сформувані навички проектування та створення баз даних, введення та редагування даних в таблиці. Навчити учнів задавати полям різні типи даних.
Базові поняття й терміни:	Таблиця, Конструктор, Майстер таблиць, поле, типи полів.

Хід уроку**1. Мотивація навчальної діяльності учнів**

На минулому уроці ми вивчали теоретичний матеріал про БД та СУБД. Сьогодні ми будемо самостійно створювати таблиці – основний атрибут сучасної СУБД.

2. Оголошення теми уроку.**3. Відтворення і корекція опорних знань учнів.**

Проводиться опитування, в результаті якого учні повинні:

- дати означення понять, що були вивчення на минулому уроці, зокрема таких як *база даних, система управління базами даних;*
- навести приклади (назви) різноманітних СУБД;
- назвати моделі баз даних;
- перелічити основні об'єкти СУБД Access;
- сформулювати означення головного об'єкта – таблиці;
- скласти логічну схему створення БД.

Під час опитування потрібно охопити якомога більшу кількість учнів, при необхідності учні коректують та доповнюють відповіді товаришів.

4. Вивчення нового матеріалу.

Матеріал уроку доцільно вивчати в такій послідовності:

- запуск MS-Access;
- ознайомлення з інтерфейсом та структурою MS-Access;
- створення таблиць;
- заповнення таблиць.

При вивченні матеріалу бажано ілюструвати розповідь демонстраціями по локальній мережі, або використовуючи стенди, плакати тощо.

Запуск MS-Access відбувається аналогічно до всіх уже відомих учням програм – з допомогою іконки на робочому столі або через кнопку "Пуск".

При запуску Access відкривається вікно, в якому можна вибрати дію: створити нову базу даних чи відкрити ту, що є на диску.



Зовнішній вигляд програми подібний до інших програм пакету Microsoft Office, які вивчалися раніше. Але є важлива відмінність – панель інструментів змінюється в залежності від того, з яким об'єктом працювати.

Таблиці – це об'єкти, в яких безпосередньо зберігаються дані.

Створити таблицю можна, вибравши у вікні БД вкладку "Таблиця" і використавши *Конструктор* або *Мастер*. але є й інші способи:

Спосіб	Опис
Режим таблиці	Виводить бланк (форму) абстрактної таблиці, яка потім може приймати конкретні форму і зміст.
Конструктор	У режимі конструктора можна безпосередньо вказати параметри всіх елементів структури таблиці.
Мастер таблиц	Дозволяє вибрати поля, що включаються в таблицю, із широкого списку зразків полів різних типів
Імпорт таблиц	Імпорт даних у таблицю з інших файлів.
Связь с таблицами	Встановлення зв'язків з даними, що зберігаються в таблицях інших БД.

Для заповнення таблиці необхідно перейти в режим *заповнення таблиці*, відкривши її.

Таблиці складаються з полів та записів. Полями називаються стовпчики, а рядки – записами. Внести запис до таблиці означає заповнити даними рядок. Щоб створити таблицю необхідно визначити її поля, типи даних цих полів, а іноді деякі додаткові властивості цих полів. Не всі дані займають в комп'ютері однаковий об'єм. Для їх компактного зберігання необхідно чітко визначити їх тип.

В Access передбачені наступні типи даних:

Текстовий – для введення тексту довжиною до 255 символів. Цей тип даних встановлюється за замовченням.

Поле МЕМО – для введення заміток або довгих описів.

Числовий – для введення числових даних.

Дата/час – для введення дати і часу.

Грошовий – використовується для роботи з грошовими величинами.

Лічильник – для введення числа, яке автоматично збільшується на одиницю при додаванні в таблицю нового запису.

Логічний – для збереження логічного значення Так або Ні.

Об'єкти OLE – використовується для збереження в таблиці OLE-об'єктів (наприклад, малюнків, звуків, документів Word та ін.).

Гіперпосилання – для запису в таблицю гіперпосилань (шлях URL).

Учням слід наголосити на принциповій різниці між роботою в режимі заповнення і роботою в режимі конструктора, а також на важливості правильного вибору типу даних.

На завершення розповіді бажано надати дітям такі поради щодо грамотного та раціонального створення таблиць:



- Інформація в таблицях не повинна дублюватися.
- Бажано, щоб кожна таблиця містила інформацію лише на одну тему.
- Не рекомендується включати до таблиці дані, що утворюються внаслідок обчислень.
- Інформацію про об'єкт бажано розбивати на мінімальні одиниці.

5. Практична робота учнів. Інструктаж з техніки безпеки.

Учням пропонується виконати завдання згідно поданого плану:

1. Завантажити БД Access.
2. В діалоговому вікні Microsoft Access створити нову базу даних. Ввести назву бази даних – "Школа - *прізвище*". (Прізвище пише кожен своє).
3. Зберегти БД "Школа ... " в вказаному вчителем місці.
4. В режимі конструктора створити таблицю Учень.

№ справи	Прізвище	Ім'я	По-батькові	Дата народження	Домашня адреса

Для полів: Прізвище, Ім'я, По-батькові, Домашня адреса, вибрати тип поля – *текстовий*, для поля № справи – *лічильник*, Дата народження – *дата/час*.

5. Перейти в режим заповнення таблиці. Ввести в таблицю дані про 5 учнів (для зручності – з цієї ж групи чи класу)
6. Закрити таблицю (зберігаючи всі зміни) і вийти із Access.
7. При необхідності, за вказівкою вчителя вимкнути комп'ютер.

6. Підведення підсумків роботи

Учні звітують про успішність виконання практичного завдання, колективно обговорюються проблеми та помилки, що виникли під час роботи.

7. Закріплення вивченого теоретичного матеріалу.

Учням пропонується дати відповіді на запитання:

- Чим відрізняється режим Конструктора від Мастера таблиць?
- Чим відрізняється автоматичний спосіб створення таблиць від ручного?

8. Домашнє завдання. Опрацювати конспект уроку.

Р.2, §5.1, 5.2, 5.4, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов,

§ 15.1, 15.3 "Курс інформатики". В.Д. Руденко, О.М. Макарчук, М.О. Патланжоглу.

8. Підведення підсумків уроку.



Урок № 18

<i>Тема:</i>	Редагування даних. Форми.
<i>Мета:</i>	Поглибити знання учнів про бази даних. Закріпити навички щодо прийомів створення і введення даних в базу даних. Ознайомити з об'єктом "Форма". Формувати практичні навички роботи в режимі редагування бази даних та створення форм.
<i>Базові поняття й терміни:</i>	Редагування БД, форма

Хід уроку

1. Мотивація навчальної діяльності учнів

Відразу створити ідеально правильну таблицю (тим більше всю базу даних) неможливо. Тому рано чи пізно настає потреба редагування даних. Окрім цього, більшість баз даних є динамічними – вони постійно оновлюються. Тому на сьогоднішньому уроці ми будемо вивчати, як можна змінювати матеріали в таблицях.

2. Оголошення теми уроку.

3. Відтворення і корекція опорних знань учнів.

Інформаційна розминка:

- Які основні етапи при створенні бази даних?
- Що потрібно зробити на етапі проектування?
- Як створити таблицю в режимі Конструктора?
- Як створити таблицю в режимі Мастера?



Самостійна робота.

Учням пропонується заповнити таблицю, вказавши призначення кожного з об'єктів СУБД Access:


<i>Об'єкт</i>	<i>Призначення</i>
Таблиця	
Запрос	
Форма	
Отчет	
Макрос	
Модуль	

4. Вивчення нового матеріалу.

Серед найпоширеніших операцій в базах можна виділити такі:

<i>Дія</i>	<i>Способи виконання</i>
Переключання між режимом конструктора і режимом таблиці	 Кнопка панелі інструментів у режимі конструктора
	 Кнопка панелі інструментів у режимі таблиці
Додавання поля в таблицю в режимі таблиці	1. Активізувати комірку в стовпці, ліворуч від якого необхідно додати новий стовпець



	2. Меню "Вставка", команда "Столбец" 3. Подвійне натискання на заголовку нового стовпця, ввести ім'я поля
Створення або зміна ключового поля	Кнопка  панелі інструментів у режимі конструктора Меню "Правка", команда "Ключове поле"
Створення зв'язків між таблицями	<i>Вивчатиметься на наступних уроках.</i>

Форми.

Розглянуті раніше способи подачі даних — табличні. Для деяких користувачів табличне подання даних може бути незручним. Часто дані необхідно розташовувати у такому вигляді, в якому зберігаються відповідні документи. Система Access дозволяє розташовувати на екрані дані так, як того бажає користувач, тобто у вигляді різноманітних форм. Форма може використовуватись також для пошуку даних.

Форма може бути створена користувачем у режимі *Конструктора форм* або за допомогою *Мастера*.

Система Access може автоматично створювати три види форм: стовпцеву, стрічкову та табличну. У відкритому вікні бази даних відкриваємо вкладку *Форми* і натискаємо кнопку *Создать форму с помощью мастера*. Далі робота буде полягати у поетапному проходженні кроків, запропонованих *Майстром*. Серед них головні: вибір об'єкта, для якого буде створюватися форма; вибір полів, що попадуть у форму; вибір вигляду форми.

Приклад готової форми:

Принципи використання форм:

- Форми можуть відображати тільки один запис за один раз, звичайно у простому форматі;
- Форми можуть відображати як поля, що можна редагувати, так і незмінні поля;
- Форми можна розробити схожими на звичайні паперові бланки;
- Форми дозволяють змінити порядок дотримання полів, щоб полегшити введення даних і зробити його більш акуратним;
- Форми можуть містити дані з декількох таблиць;



- Форми можуть містити графіки і діаграми;
- Форми дозволяють автоматизувати задачі введення і створювати меню користувача.

5. Практична робота учнів. Інструктаж з техніки безпеки.

Редагування:

1. Завантажити БД Access.
2. Відкрити базу даних "Школа ... "
3. Відкрити таблицю "Учень" в режимі конструктора.
4. Попереду поля *Дата народження* додайте поле *Клас* (тип поля – текстовий, розмір поля – 4 символи).
5. Зробіть заміну поля *Домашня адреса* на *Місце проживання* (Перейменуйте).
6. Додайте поле *Додаткова інформація* (тип поля – текстовий, розмір – 50).
7. Перейдіть в режим таблиці.
8. Перетягніть поле *Клас* в кінець таблиці.
9. Додайте в таблицю 2 нових записи.
10. Закрити таблицю (зберігаючи всі зміни).

Створення форм:

1. Активізувати вкладку "Формы".
2. Запустити Майстер створення форм.
3. Створити форму "Учень", взявши всі поля з таблиці "Учень"
4. Перевірити роботу форми, переглянувши дані полів.
5. Ввести 1 новий запис до таблиці, використовуючи створену форму.
6. Закрити форму.
7. Вийти із Access.
8. При необхідності, за вказівкою вчителя вимкнути комп'ютер.

6. Підведення підсумків роботи

Учні звітують про успішність виконання практичного завдання, колективно обговорюються проблеми та помилки, що виникли під час роботи.

7. Закріплення вивченого теоретичного матеріалу.

Учням пропонується дата відповіді на запитання:

- Для чого потрібна структура Форми?
- Які елементи можна вставити в форму в режимі конструктора?
- Як у формі реалізовується перехід між записами?
- Які недоліки можуть мати форми?

8. Домашнє завдання. Опрацювати конспект уроку.

Р.2, §5.4, 5.8 Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов,
§ 15.2, 15.4, 15.7 "Курс інформатики". В.Д. Руденко, О.М. Макарчук, М.О. Патланжоглу.

9. Підведення підсумків уроку.

**Урок № 19**

<i>Тема:</i>	Поля підстановки. Створення зв'язків між таблицями. Ключові поля.
<i>Мета:</i>	Закріпити практичні навички учнів роботи з базами даних. Навчити встановлювати відношення між таблицями та створювати зв'язки. Сформувані практичні навички створення поля підстановки.
<i>Базові поняття й терміни:</i>	Поле підстановки, ключове поле, зв'язки.

Хід уроку


1. Оголошення теми уроку.
2. Вивчення нового матеріалу.

Розповідь вчителя з використанням демонстрації по локальній мережі.

Відношення між таблицями

Можливі відношення між таблицями бази даних бувають таких типів: "один до одного", "один до багатьох", "багато до одного" і "багато до багатьох". Найпоширенішим у таблицях реляційних баз даних є відношення "один до багатьох".

Відношення "один до багатьох" означає, що одному запису таблиці відповідають кілька записів в іншій таблиці

Починаючи зв'язування таблиць, переконайтеся, що всі таблиці і форми закриті. Клацніть по кнопці *Схема даних*  на панелі інструментів у відповідь відкриється однойменне діалогове вікно.

Водночас з вікном *Схема даних* з'явиться діалог *Добавление таблицы*. Однак цей діалог може і не з'явитися, якщо раніше були створені зв'язки між таблицями (вікно *Схема даних* не порожнє). У цьому разі потрібно очистити вікно від зв'язків, клацнувши по кнопці *Очистить макет*. Потім натисніть кнопку *Добавить таблицу* на панелі інструментів.

За замовчуванням в діалозі *Добавление таблицы* буде відкрита вкладка *Таблицы*. Потрібно виділити у списку потрібну таблицю, і натисніть кнопку *Добавить*. У вікні *Схема даних* з'явиться список полів таблиці.

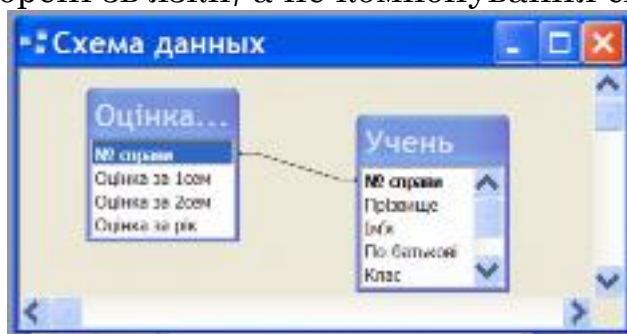
Аналогічно виводяться списки полів інших таблиць у вікно *Схема даних*, після чого потрібно закрити діалог *Добавление таблицы*.

Для зв'язування полів оберіть поле у головній таблиці і перетягніть його мишею до зв'язаної таблиці. Слід зазначити, що напрямок перетягування поля завжди повинен бути «від головної таблиці до зв'язаної».



Макет зв'язків

Списки полів у вікні *Схема даних* розташовуються зручним способом (наприклад, щоб лінії зв'язків не перетиналися) Списки можна перетягнути, захопивши мишею заголовок списку. Розташування списків у вікні називається макетом зв'язків. Після цього можна закрити вікно *Схема даних*. Програма виведе на екран запит щодо того, чи потрібно зберігати макет зв'язків. Потрібно відповісти ствердно. Якщо ви виберете "Ні", то збережуться лише створені зв'язки, а не компонування списків.



В розглянутих випадках ми зв'язували ключове поле (позначене напівжирним шрифтом у списку полів) головної таблиці з відповідним йому полем зв'язаної таблиці. Поле зв'язаної таблиці називають полем зовнішнього ключа.

Для видалення будь-якого зв'язку не потрібно знову створювати макет зв'язків. Виділіть зв'язок клацанням миші й натисніть клавішу Delete. Після появи запиту на видалення зв'язку клацніть по кнопці "Так".

Поле підстановки

До зв'язування таблиць також відноситься і "поле підстановки".

Поле підстановки дає можливість при заповненні таблиці використовувати дані з інших таблиць, що значно економить час і зусилля.

Для використання поля підстановки потрібно:

- в стовпці *Тип даних* із списку даних вибрати *мастер підстановок*.
- пройти всі етапи *Мастера підстановок*, потрібно буде вказати таблицю, з якої будуть братися дані, а також поле в цій таблиці.
- початкову таблицю відкрити в режимі заповнення, і в відповідному вікні для заповнення досить буде вибрати з випадаючого списку потрібні дані.

3. Практична робота учнів. Інструктаж з техніки безпеки.

Робота з полями підстановки

- Запустіть Access. Відкрити БД "Школа".
- Створіть і збережіть таблицю "Секції" в своїй базі даних "Школа ... "

Код секції	Назва секції
1	плавання
2	гімнастика
3	бокс
4	теніс
5	спортивні танці




- Збережіть та закрийте цю таблицю.
- Відкрийте таблицю "Учень" в режимі конструктора.
- Створіть нове поле "Секція", організуйте в це поле підстановку даних з поля "Назва секції" таблиці "Секції"
- Відкрийте таблицю "Учень" в режимі таблиці.
- В стовпці "Секція" із розгорнутого списку введіть значення для всіх рядків.

Створення зв'язків між таблицями.

- Створити таблицю "Оцінка з інформатики".

№ справи	Оцінка за I сем.	Оцінка за II сем.	Оцінка за рік.

- За допомогою кнопки *Схема данных*  розташованій на панелі інструментів, відкрити вікно для створення зв'язків.
- Виберіть таблицю "Учень" і додайте її в нову схему даних.
- Додайте в схему даних таблицю "Оцінка з інформатики".
- Для створення зв'язку перетягніть поле № справи із таблиці "Учень" в таблицю "Оцінка з інформатики". В результаті з'явиться лінія, яка з'єднує дві таблиці.
- Закрийте таблицю "Учень", збережіть зміни.
- Перевірте наявність зв'язків, відкривши таблицю "Учень" в режимі заповнення
- Закрийте таблицю і вийти із Access.

4. Підведення підсумків роботи

Учні звітують про успішність виконання практичного завдання, колективно обговорюються проблеми та помилки, що виникли під час роботи.

5. Закріплення вивченого теоретичного матеріалу.

Учням пропонується дати відповіді на запитання, при цьому дозволивши їм обговорення попарно (в межах парті):

- Які відношення між таблицями вам відомі?
- Що таке головна і зв'язана таблиці у відношенні «один до багатьох»?
- Опишіть процедуру створення зв'язків між таблицями.
- Як виглядатиме лінія зв'язку у вікні *Схема данных*, якщо не установити перемикач *Обеспечение целостности данных*?
- Що таке макет зв'язків?
- Як видалити зв'язок між таблицями?
- Що таке ключове поле?

6. Домашнє завдання. Опрацювати конспект уроку.

Р.2, §5.3, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов, § 15.5 "Курс інформатики". В.Д. Руденко, О.М. Макарчук, М.О. Патланжоглу.

7. Підведення підсумків уроку.



Урок № 20

Тема:	Сортування, фільтрація і пошук даних.
Мета:	Закріпити практичні навички учнів роботи в режимі редагування бази даних. Ознайомити учнів з методами пошуку, сортування та фільтрації даних в БД. Сформувати практичні навички пошуку, сортування та фільтрації даних в БД.
Базові поняття й терміни:	Сортування, фільтрація, пошук даних, розширений фільтр, автофільтр.

Хід уроку

1. Організаційний момент
2. Оголошення теми уроку.
3. Відтворення і корекція опорних знань учнів.

Експрес-опитування

- Що таке ключове поле?
- Опишіть процедуру створення зв'язків між таблицями.
- Що таке макет зв'язків?
- Як видалити зв'язок між таблицями?
- Які відношення між таблицями вам відомі?

4. Вивчення нового матеріалу.

а) Аналіз таблиці.



Сортування і фільтрація даних

Дія	Спосіб виконання
Сортування даних у полі	Контекстне меню, вибрати <i>Сортування по...</i>
	Кнопки  панелі інструментів у режимі таблиці
Накладення фільтра на записи згідно з виділеним фрагментом	Контекстне меню, вибрати <i>Фільтр по виділеному</i>
	Кнопка  панелі інструментів у режимі таблиці
Зміна критеріїв фільтрації	Контекстне меню заголовка таблиці, вибрати <i>Змінити фільтр</i>
	Кнопка  панелі інструментів у режимі таблиці
Застосування фільтра	Контекстне меню заголовка таблиці, вибрати <i>Застосувати фільтр</i>
	Кнопка  панелі інструментів у режимі таблиці
Вилучення фільтра (відновлення вмісту таблиці)	Контекстне меню заголовка таблиці, вибрати <i>Видалити фільтр</i>
	Кнопка  панелі інструментів у режимі таблиці



5. Практична робота учнів. Інструктаж з техніки безпеки.

Учням пропонується виконати завдання згідно поданого плану:

- Активізуйте вкладку *Таблицы*. Відкрийте таблицю "Учень".
- На панелі інструментів натисніть кнопку 
- В діалоговому вікні *Поиск в поле* у полі *Образец* введіть значення.
- Натисніть кнопку *Найти*. У формі з'явиться запис, відповідаючий умові пошуку.
- Для пошуку наступних записів натисніть кнопку *Найти далее*.
- Натисніть кнопку *Закреть* у діалоговому вікні *Поиск в поле*.
- Застосуйте фільтр у полі *По-батькові*, для записів, "Іванович", "Іванівна" (Для цього клацаємо в будь-якій комірці поля *По-батькові*, яка має відповідне значення. На панелі інструментів натискаємо кнопку  (*Фильтр по выделенному*)
- Продемонструйте результати вчителю.
- Відмініть фільтр.
- Відсортуйте прізвища за алфавітом.
- Збережіть зміни. Закрийте таблицю "Учень".
- Закрийте таблицю і завершіть роботу з Access.

6. Підведення підсумків роботи

Учні звітують про успішність виконання практичного завдання, колективно обговорюються проблеми та помилки, що виникли під час роботи.

7. Закріплення вивченого теоретичного матеріалу.

Учням пропонується дати відповіді на запитання:

- Для чого потрібні фільтри?
- Чи зберігаються автоматично при виході з таблиці замінені записи?

8. Домашнє завдання. Опрацювати конспект уроку.

9. Підведення підсумків уроку.



Урок № 21

<i>Тема:</i>	Запити
<i>Мета:</i>	Продовжити ознайомлювати учнів з БД Access. Сформувати вміння працювати із запитам.
<i>Базові поняття й терміни:</i>	Запити, операнди і ідентифікатори, вибірка.

Хід уроку

1. Мотивація навчальної діяльності учнів

Запит у системі Access використовують для пошуку з однієї чи кількох таблиць та видачі на екран даних, які відповідають умові, визначеній користувачем.

2. Оголошення теми уроку.

3. Відтворення і корекція опорних знань учнів.

Учитель проводить опитування, в результаті якого учні повинні дати відповіді на запитання:

- Для чого потрібні фільтри?
- Чи зберігаються автоматично при виході з таблиці змінені записи?

4. Вивчення нового матеріалу.

а) Шкільна лекція:

Запити – це умови, за допомогою яких можна скласти набір необхідних полів (у тому числі і з декількох таблиць) і задати такі значення полів, за якими вибираються записи для обробки, тобто задати фільтр відбору інформації.

Через запит можна робити поновлення даних в таблицях, доповнення та вилучення записів. Запит будується на основі однієї або декількох таблиць бази даних. Крім того, можуть використовуватися збережені таблиці, отримані в результаті виконання інших запитів. Запит може будуватися безпосередньо на іншому запиті з використанням його тимчасової таблиці з результатами.

У Microsoft Access є зручний для користувача графічний засіб формування запиту за зразком, за допомогою якого легко може бути побудований складний запит. При конструюванні запиту достатньо, працюючи мишею, виділити і перемістити необхідні поля з таблиць, поданих у схемі даних запиту, у бланк запиту і ввести умови вибору записів.

Призначення запитів

Засобами запиту можна:

- вибрати записи, що задовольняють умовам відбору;
- включити в результуючу таблицю запиту потрібні поля;
- зробити обчислення для кожного з отриманих записів;



- згрупувати записи з однаковими значеннями в одному або декількох полях і виконати для них групові функції;
- зробити поновлення полів в обраній підмножині записів;
- створити нову таблицю бази даних, використовуючи дані з існуючих таблиць;
- вилучити обрану підмножину записів із таблиці бази даних;
- додати деяку підмножину записів до іншої таблиці.

Багатотабличний запит дозволяє сформуванню нову таблицю, записи якої утворюються шляхом об'єднання взаємозалежних записів з різних таблиць бази даних і розміщення потрібних полів з цих таблиць.

Послідовне виконання ряду запитів дозволяє вирішувати досить складні задачі, не застосовуючи програмування.

б) Робота з таблицею "Види запитів"

В Access може бути створено декілька видів запитів:

Тип запиту	Опис
Запити на вибірку	Дані, що задовольняють умовам відбору, вибираються з однієї або декількох таблиць і виводяться в певному порядку.
Запити з параметрами	При створенні запиту необхідно у вікні діалогу ввести певні відомості, наприклад, умови відбору записів або значення для вставки в поле. Зручно використати як базове джерело даних для форм або звітів.
Перехресні запити	Відображають результати статистичних розрахунків, виконаних за даними з одного поля. Ці результати групуються по двох наборах даних у форматі перехресної таблиці, утворюючи номери рядків і заголовки стовпців
Запити на зміну	Дозволяють в одній операції внести зміни у багато записів. Існує чотири типи записів на зміну: на створення таблиці, на видалення записів, на додавання і на оновлення записів
Запити SQL	Запитом SQL називають запит, що створюється за допомогою інструкції SQL (мови структурованих запитів). Прикладами запитів SQL є запит на об'єднання, запит до сервера, керуючий запит та ін.

Умови відбору записів

Умови відбору записів можуть задаватися для одного або декількох полів у відповідному рядку бланка запиту. Умовою відбору є вираз, що складається з операторів порівняння та операндів і ідентифікаторів, які використовуються для порівняння.

Ідентифікатор є посиланням на значення поля, елемента керування або властивості. Ідентифікаторами можуть бути імена полів, таблиць, запитів, форм, звітів і т.д. Вони повинні розміщуватись в квадратних дужках. Якщо необхідно задати посилання на поле в конкретній таблиці (формі, звіті), то перед іменем поля також в квадратних дужках ставиться ім'я таблиці, відокремлене від імені поля знаком оклику.



Слід пам'ятати, що *Запрос* - це тимчасова таблиця. Це означає, що дані в них не зберігаються постійно, а тільки тимчасово викликаються з таблиці, по заданому шаблону, в момент активізації запиту. Таким чином, в базі даних постійно зберігається тільки шаблон виклику даних (тимчасові таблиці видаляються після закриття запиту), а сама інформація не дублюється.

5. Виконання практичного завдання.

Учням пропонується виконати завдання згідно поданого плану:

1. Відкрити БД "Школа..."
2. Перейдіть на вкладку *Запрос*.
3. За допомогою Конструктора створіть запит, використовуючи таблицю *Учень*, для запиту вибрати такі стовпці: Прізвище, Ім'я, По батькові, Секція. В запиті необхідно вказати: - *сортировку по возрастанию*; для поля *Секция* визначить *условие отбора* - гімнастика. В результаті виконання запиту з'явиться відсортована в алфавітному порядку таблиця учнів, які відвідують гімнастику.
4. Збережіть запит на закрийте його.

6. Підведення підсумків роботи

Учні звітують про успішність виконання практичного завдання, колективно обговорюються проблеми та помилки, що виникли під час роботи.

7. Формування вміння самостійно здобувати знання.

Самостійна робота (знайти відповідь на питання в довідковій системі СУБД):

- Для чого використовуються запити в базах даних?
- За допомогою чого і як створюються запити?
- Як зробити запит за визначеним параметром (вибірку)?
- Як створюються підсумкові запити?

8. Домашнє завдання. Опрацювати конспект уроку.

Р.2, §5.6, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов, §15.6 "Курс інформатики". В.Д. Руденко, О.М. Макаруч, М.О. Патланжоглу.

9. Підведення підсумків уроку.

**Урок № 22****Тема:** Звіти.**Мета:** Продовжити ознайомлювати учнів з БД Access, формувати вміння створювати звіти.**Базові поняття й терміни:** Звіт, Конструктор звіту, заголовок, колонтитули, область даних.**Хід уроку**

1. **Організаційний момент**
2. **Оголошення теми та мети уроку.**
3. **Відтворення і корекція опорних знань учнів.**

Опитування:

- Для чого використовуються запити в базах даних?
- За допомогою чого і як створюються запити?
- Як зробити запит за визначеним параметром (вибірку)?
- Як створюються підсумкові запити?

4. Вивчення нового матеріалу.

а) Розповідь вчителя з елементами демонстрації по локальній мережі.

Звіт – це засіб створення і друку підсумкових документів за наявною у БД інформацією. Звіт може бути створений на основі таблиць або запитів.

Складові частини звіту:

Заголовок. Інформація на початку першої сторінки (назва).

Верхній колонтитул. Інформація на початку кожної сторінки (заголовки стовпців таблиць).

Область даних. Відображення даних із таблиць або запитів.

Примітка групи. Інформація в кінці групи даних (підсумок за групою).

Нижній колонтитул. Інформація в кінці кожної сторінки (номер сторінки).

Область приміток звіту. Інформація в кінці останньої сторінки (підсумкові обчислення по всім записам звіту).

Прикладу звіту зробленого *Мастером*:

Зарплата	
Прізвище	Зарплата
Петренко А.І.	1075.8
Івасюк П.А.	1243.2
Кириленко Л.П.	1164.4
Безрученко І.І.	873.2
Зінченко В.Г.	1016.4
Чумак А.В.	1095.2
Гусаченко Є.М.	1443.2
Балтак А.П.	805.2
Береговий С.Я.	1058.2



Формування і виведення звітів

Засоби Access по розробці звітів призначені для створення звіту, за яким може бути здійснене виведення даних із таблиць у вигляді результуючого друкарського документа. Ці засоби дозволяють конструювати звіт складної структури, що забезпечує виведення взаємозалежних даних із багатьох таблиць, при цьому можуть бути реалізовані найвищі вимоги до оформлення документа, перед початком конструювання звіту користувач повинен зробити підготовчу роботу, у результаті якої потрібно визначити необхідний макет звіту.

У процесі конструювання визначається склад та зміст розділів звіту, а також розміщення в ньому значень полів таблиць бази даних. Крім того, оформляються заголовки, підписи реквізитів звіту, визначається розміщення обчислюваних реквізитів.

Засоби конструювання звіту дозволяють групувати дані на декількох рівнях. Для кожного рівня можуть провадитися обчислення результатів, призначатися заголовки і примітки по кожному угрупованню. При формуванні звіту можуть виконуватися різноманітні обчислення.

Звіт може створюватися за допомогою *Мастера* або в режимі *Конструктора отчетов*. У багатьох випадках зручно використовувати *Мастера*, створений *Мастером* звіт можна дооформити в режимі *Конструктора*.

При необхідності виведення в звіті даних із багатьох таблиць за основу для звіту може бути використаний багатотабличний запит. На запит можуть бути покладені найбільш складні види вибірки і попередньої обробки даних, Конструктор звітів дозволяє отримані в запиті дані успішно структурувати і оформляти.

б) Самостійна робота учнів.

Опрацювати теоретичний матеріал (використавши довідкову систему СУБД, написати короткий конспект)

Перегляд і друк звіту

Для перегляду раніше створеного звіту потрібно у вікні бази даних вибрати "Отчеты" і натиснути кнопку "Просмотр". Звіт при перегляді з'явиться на екрані таким, яким він буде надрукований. Перехід у режим попереднього перегляду з режиму *Конструктора отчетов* виконується натисканням кнопки "Представление отчета". В режимі попереднього перегляду є своя панель інструментів.

Кнопка "Печать" панелі інструментів режиму попереднього перегляду дозволяє вивести даний звіт на друк. За допомогою команди "Файл-Макет страницы" можна вибрати принтер, задати формат паперу, розмір полів, відстань між рядками, орієнтацію і т.д. Команда "Файл-Печать" дозволяє вибрати для друку окремі сторінки звіту або виділені записи, роздрукувати декілька копій, вивести звіт у файл.



Звіт багато в чому схожий на форму. Його використовують при виведенні документів на друк. Принципова різниця між Формою та Отчетом у тому, що формати звітів відповідають стандартним форматам паперу.

5. Виконання практичного завдання.

План виконання практичного завдання:

- Перейдіть на вкладку *Отчет*. Створіть звіт за допомогою *Мастера отчетов*, використовуючи створений на минулому уроці запит.
- Відкрийте запит в режимі *Конструктора*. Зробіть заголовок "Звіт до завдання".
- Збережіть звіт та закрийте його.
- При необхідності, за вказівкою вчителя вимкніть комп'ютер.

6. Підведення підсумків роботи

Учні звітують про успішність виконання практичного завдання, колективно обговорюються проблеми та помилки, що виникли під час роботи.

7. Закріплення вивченого теоретичного матеріалу.

Учням пропонується дата відповіді на запитання, при цьому дозволивши їм обговорення попарно (в межах парти):

- Для чого використовуються звіти в базах даних?
- За допомогою чого і як створюються звіти?

8. Домашнє завдання. Опрацювати конспект уроку.

Р.2, §5.8, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов, § 15.6 "Курс інформатики". В.Д. Руденко, О.М. Макаруч, М.О. Патланжоглу.

Підготуватись до ТО по темі "Бази даних. Системи управління базами даних".

9. Підведення підсумків уроку.



Урок № 23

<i>Тема:</i>	Тематичне оцінювання з теми "Бази даних. Системи управління базами даних".
<i>Мета:</i>	Перевірка та контроль знань, вмінь і навичок учнів.

Хід уроку

1. Організаційний етап.
2. Оголошення теми і мети уроку.
3. Виконання завдань тематичного оцінювання.

Варіант I проведення тематичного оцінювання.

1. Письмова самостійна робота.

Варіант I

1. Назвіть, які операції з даними можна виконувати в БД? (1 бал)
2. Яка структура екрана в СУБД? (1 бал)
3. Чи є основним призначенням БД створення і редагування текстових документів? (1 бал)
4. Які етапи створення БД? (2 бали)
5. Що таке поле, запис, файл БД? (2 бали)
6. Який тип потрібно встановити для поля „Прізвище учня” - числовий чи текстовий? (2 бали)
7. В якому режимі виконується заповнення, редагування та пошук інформації в БД? (2 бали)
8. Які типи даних використовують в БД? (3 бали)
9. Що таке „запит”, його види? (3 бали)
10. Чи може в ключовому полі бути декілька однакових значень у різних записах? (3 бали)
11. Основні принципи захисту інформації. (4 бали)
12. Модулі і їх призначення. (4 бали)

Варіант II

1. Яка інформація може бути записана в БД? (1 бал)
2. Чи можливо в БД здійснити пошук інформації за деякою ознакою? (1 бал)
3. Чи дозволяє СУБД виконувати розрахунки за формулами? (1 бал)
4. Основне призначення БД і СУБД. (2 бали)
5. Які обов'язкові характеристики повинен вказати користувач для поля? (2 бали)
6. Що таке сортування інформації, які її види? (2 бали)
7. Який тип потрібно встановити для поля „Кількість людей в класі” – текстовий чи числовий? (2 бали)
8. В якому режимі створюється структура БД? (3 бали)
9. Що таке „звіт”, його види? (3 бали)



10. Якщо не вказати ключове поле при збереженні БД, то яку дію виконає програма самостійно? (3 бали)
11. Як створити структуру БД? (4 бали)
12. Макроси і їх призначення. (4 бали)

Бали	1	2	3	4-5	6-7	8-9	10-11	12-14	14-17	18-20	21-24	25-28
Оцінка	1	2	3	4	5	6	7	8	9	10	11	12

Варіант II тематичного оцінювання ("Експрес-тест").

1. База даних призначена для:
 - а) зберігання і впорядкування інформації;
 - б) ведення розрахунково-обчислювальних операцій;
 - в) обробки текстової інформації;
 - г) обробки графічної інформації.
2. Текстове поле дозволяє занести інформацію до:
 - а) 20 символів;
 - б) 1 символу;
 - в) 256 символів;
 - г) 65536 символів.
3. Записами вважаються:
 - а) заголовки;
 - б) рядки;
 - в) стовпці;
 - г) таблиці.
4. Таблиці служать для:
 - а) ведення розрахунків;
 - б) збереження інформації
 - в) вибірки інформації;
 - г) виводу на друк.
5. Кнопка „Конструктор” відкриває:
 - а) вміст таблиці;
 - б) структуру об'єкта;
 - в) панель елементів;
 - г) немає вірної відповіді.
6. Зв'язки між таблицями потрібні для:
 - а) копіювання даних;
 - б) забезпечення цілісності даних;
 - в) створення звітів, записів;
 - г) немає вірної відповіді.
7. Для створення запиту потрібно виконати:
 - а) *Запросы* → *Создать ...*
 - б) *Вставка* → *Запрос ...*
 - в) *Вставка* → *Автозапрос ...*
 - г) *Сервис* → *Параметры ...*
8. Звіти створюються для:



- а) введення даних;
 - б) виводу даних на друк;
 - в) вибірки інформації;
 - г) копіювання даних.
9. Як називаються інформаційні структуровані одиниці, які складаються з полів:
- а) файли даних;
 - б) бази даних;
 - в) поля;
 - г) записи.
10. Яку команду використовують для відбору записів за виділеним зразком:
- а) удалить фільтр;
 - б) фільтр по выделенному;
 - в) исключить выделенное;
 - г) объект.
11. За допомогою якої команди виділені записи розміщуються в буфері обміну, при цьому залишаючись в самій таблиці:
- а) вирізати;
 - б) видалити;
 - в) копіювати;
 - г) вставити.
12. Для оформлення потрібних даних з бази даних згідно з вимогами стандартів чи замовника і виведення їх на папір призначені:
- а) форми;
 - б) звіти;
 - в) запити;
 - г) фільтри.
13. Для чого використовують фільтри в БД?
- а) для пошуку даних;
 - б) для виведення даних на екран;
 - в) для виведення даних на папір;
 - г) для збереження даних на диску.
14. Як називається перехрещення поля і запису в БД?
- а) комірка;
 - б) комірка;
 - в) гніздо;
 - г) значок.
15. Структура таблиці в БД Access це...
- а) сукупність полів;
 - б) структура запису;
 - в) сукупність типів;
 - г) сукупність властивостей.



16. Який тип даних використовують в БД Access для збереження будь-якої послідовності символів?
 - а) текстовий;
 - б) МЕМО;
 - в) лічильник;
 - г) числовий.
17. Який об'єкт БД дозволяє відображати інформацію в зручному для користувача вигляді?
 - а) таблиці;
 - б) запити;
 - в) форми;
 - г) зв'язки.
18. Який тип зв'язку встановлюється при зв'язуванні двох ключових полів?
 - а) один до одного;
 - б) кілька до кількох;
 - в) один до кількох;
 - г) невизначений.
19. Що треба зробити для створення БД в першу чергу?
 - а) створити її на папері;
 - б) визначити вхідні дані;
 - в) визначити тип даних;
 - г) перелічити всі типи даних.
20. Якого типу даних немає для числового поля?
 - а) лічильник;
 - б) ціле;
 - в) байт;
 - г) довге ціле.
21. Як називається рядок таблиці в СУБД Access?
 - а) запис;
 - б) форма;
 - в) поле;
 - г) комірка.
22. Як називається режим роботи, в якому задається структура таблиці?
 - а) майстер таблиць;
 - б) конструктор;
 - в) художник;
 - г) майстер кнопок.

3. Практичне завдання. Інструктаж з техніки безпеки.

Варіант І

1. Запустіть програму для опрацювання бази даних Access. (1 бал)
2. Створіть нову базу даних з іменем файлу – вашим прізвищем. (1 бал)



3. Створіть структуру бази даних у вигляді таблиці „Мої однокласники” і заповніть її 5-ма записами за зразком : (2 бали)

Номер	Прізвище учня	Ім'я учня	Дата народження (число, місяць, рік)	Телефон
1	Іванов	Стас	12.05.1990	24367
2	Кирилюк	Наталія	4.11.1990	23649
3	Павлюк	Руслана	30.06.1990	24316
4	Синиця	Павло	21.11.1990	24384
5	Кубайчук	Віталій	7.10.1989	23756

4. Впорядкуйте записи в другому полі за алфавітом. (1 бал)
 5. Замініть у 3-му записі значення четвертого поля на 29.11.1989 (1 бал)
 6. Видаліть поле „Телефон” в таблиці „Мої однокласники” (1 бал)
 7. Відфільтруйте записи про учнів, які народилися в 11-му місяці (1 бал)
 8. Створіть стрічкову форму для даної таблиці. (1 бал)
 9. Створіть в БД з вашим прізвищем ще одну таблицю „Успішність” з слідуною структурою: номер, алгебра, фізика, інформатика, укр.. мова. Заповнити таблицю даними. (1 бал)
 10. Створіть зв'язок між таблицями „Мої однокласники” і „Успішність” з встановленим можливим режимом: Мої однокласники/номер → Успішність/номер. (1 бал)
 11. Створіть запит з параметром для шукання записів про учнів, які мають деяку, задану параметром, оцінку з інформатики. (1 бал)

Варіант II

1. Запустіть програму для опрацювання бази даних Access. (1 бал)
 2. Створіть нову базу даних з іменем файлу – вашим прізвищем. (1 бал)
 3. Створіть структуру бази даних у вигляді таблиці „Склад” і заповніть її 5-ма записами за зразком: (2 бали)

Код товару	Найменування	Кількість, кг	Ціна за 1кг	Імпорт
134	Цукор	1000	3,00	Ні
769	Борошно	978	1,95	Ні
452	Рис	30	2,30	Так
96	Макарони	69	1,46	Ні
235	Печиво	150	6,90	Ні

4. Впорядкуйте записи в другому полі за алфавітом. (1 бал)
 5. Замініть у 1-му і 4-му записі значення другого поля на цукерки (1 бал)
 6. Видаліть поле „Імпорт” в таблиці „Склад” (1 бал)
 7. Відфільтруйте записи про всі товари, крім цукерок на складі. (1 бал)
 8. Створіть стовпцеву форму для даної таблиці. (1 бал)



9. Створіть в БД з вашим прізвищем ще одну таблицю „Магазин” з слідуною структурою: код магазину (ключове поле), дата отримання, код товару, кількість отриманого товару, експедитор. Заповнити таблицю 5-ма записами. (1 бал)
10. Створіть зв'язок між таблицями „Склад” і „Магазин” з встановленим можливим режимом: Склад/код товару → магазин/код товару. (1 бал)
11. Створіть запит: вивести номери магазинів, які отримали товар в листопаді, з вказаною назвою товару і його кількість. (1 бал)

4. Домашнє завдання.

Повт. Р.2, §1-5 Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов,

§ 13, 15 "Курс інформатики". В.Д. Руденко, О.М. Макарчук, М.О. Патланжоглу.



Урок № 24

Тема: Поняття інформаційної моделі. Поняття технології комп'ютерного моделювання. Інформаційні моделі. Побудова моделі. Основні етапи комп'ютерного моделювання.

Мета: Ввести поняття математичної, інформаційної та комп'ютерної моделей.
Навчити будувати математичні моделі прикладних задач.
Вивчити основні етапи розв'язування прикладної задачі з використанням ПК.

Базові поняття й терміни: Математична, інформаційна та комп'ютерна модель. Комп'ютерне моделювання.

Хід уроку

1. Організаційний етап

Ви починаєте вивчати новий розділ інформатики – «Основи алгоритмізації та програмування. Комп'ютерне моделювання». Тема розрахована на 12 годин. Під час вивчення теми ви навчитесь створювати різноманітні алгоритми та програми мовою програмування "Turbo Pascal", виконаєте ряд практичних завдань.

Учні ознайомити з:

- критеріями оцінювання рівня навчальних досягнень.
- орієнтовним текстом роботи для тематичного оцінювання.

2. Оголошення теми уроку. Мотивація навчальної діяльності.

Розв'язування будь-яких задач - це завжди отримання певних результатів, які можуть бути виражені в вигляді чисел, графіків тощо.

ПК є універсальним пристроєм для розв'язування різноманітних навчальних і виробничих задач. Їх універсальність полягає в можливості виконання практично всіх алгоритмів розв'язування задач у будь-якій предметній області. Необхідним для цього стає опис постановок завдань та методів їх розв'язку мовою математики з наступним записом у вигляді програм.

3. Вивчення нового матеріалу.

а) Розповідь вчителя.

Ознайомлення з алгоритмом розв'язування задачі за допомогою ПК:

- 1) розробка математичної моделі (постановка задачі);
- 2) вибір методу розв'язування задачі;
- 3) складання сценарію роботи з ПК;



- 4) побудова алгоритму розв'язування задачі;
- 5) запис алгоритму засобами мови програмування;
- 6) налагодження та тестування програми (введення програми в комп'ютер та перевірка правильності результатів її виконання);
- 7) виконання програми та аналіз отриманих результатів.

Математична модель - це наближений опис об'єкта на мові математики. Побудову математичної моделі, як правило, виконують у 3 етапи:

1. Визначають дані, які потрібні для побудови математичної моделі (аргументи).
2. Визначають, що необхідно знайти (результат) і які обмеження накладаються на результат.
3. Встановлюють зв'язки між аргументами і результатом у вигляді формул, рівнянь, нерівностей, тобто будують власне математичну модель.

Математична модель дозволяє звести розв'язування реальної задачі до розв'язування задачі математичної.

На другому етапі необхідно не тільки вибрати спосіб отримання результатів, а й вибрати оптимальний метод розв'язування задачі, який би давав найбільш правильну відповідь. Правильність розв'язку задачі перш за все залежить від правильності вибраного методу розв'язування.

Для написання сценарію програми необхідно продумати правила введення даних в ПК, форму подання інформації користувачам, реакцію комп'ютера на вказівки і дані, які вводяться. Тобто сценарій визначає правила роботи користувача з ПК.

Побудова алгоритму - наступний етап розв'язування задачі. Перед розробкою алгоритму потрібно вирішити питання про вибір структур даних для подання вхідної інформації та результатів виконання, встановити зв'язки між елементами даних, вибрати методи їх обробки, тобто створити *інформаційну модель задачі*. На цьому етапі крім творчих завдань, розв'язуються і технологічні питання.

Алгоритм, призначений для виконання на комп'ютері, повинен бути записаний мовою програмування. Розробкою програми завершуються етапи розв'язування задачі, що виконуються користувачем без використання ПК. Решту етапів виконують з використанням комп'ютера.

Перед використанням програми проводять її налагодження та тестування. Тестування полягає в тому, що користувач готує систему тестів, за допомогою якої перевіряється робота програми в різних режимах. Кожний тест містить набір вхідних даних, для яких відомий результат. Це дає змогу уникнути так званих "логічних" помилок в структурі програми.

Після закінчення тестування програми вводять вхідні дані з наступним виводом результатів на екран або принтер. На основі отриманих ре-



зультатів будують таблиці, графіки, аналізують і пояснюють у термінах вихідних даних задачі.

Поняття інформаційної (математичної) моделі.

Набір даних, що містить всю необхідну і достатню інформацію про досліджувані об'єкти чи процеси називають інформаційною моделлю.

Інформаційна модель, яка містить усі необхідні дані для розв'язання певної задачі і поміщена в пам'ять комп'ютера називається *комп'ютерною моделлю*.

Успішно розв'язувати задачі можна тільки при чіткому і водночас однозначному визначенні вимог до кінцевих результатів. Розпливчастість і невизначеність формулювань може привести до різного тлумачення умов і, як наслідок - розбіжності в оцінці правильності результатів, тобто отримання різних результатів при розв'язуванні задачі різними виконавцями.

Математична постановка задачі (формалізація) - це вимоги (до розв'язку задачі та результатів), що формулюються за допомогою математики. Математичні задачі є формалізованими з самого початку.

Для запису їх постановок використовується наступна форма запису:

Задача:	<Змістовне формулювання >
Дано:	<Перелік початкових даних >
Потрібно:	<Перелік того, що потрібно знайти >
Зв'язок:	<Зв'язок між шуканими і початковими даними >
При:	<Умови допустимості для початкових даних та результатів >

Та чи інша постановка задачі завжди ґрунтується на конкретному, змістовному формулюванні умови задачі. В ньому повинні бути виділені і перелічені всі вихідні дані і потрібні результати. У математичних постановках задач у розділах "Дано" і "Потрібно" вказують не тільки назви цих величин і об'єктів, а й їхні позначення, що надалі використовуються в описах умов допустимості початкових даних і встановлення зв'язку між ними і шуканими даними.

В окремих задачах залежність між вхідними і вихідними даними, а також умови допустимості можуть виражатися не системами рівнянь, а системами тверджень. Запис таких тверджень виконується мовою математичної логіки. Це - перш за все відомості про об'єкти та їхні властивості. Крім цього, твердження можуть вводитися через раніше введені висловлення.

"Зв'язок"- це система рівнянь або тверджень, що зв'язують вихідні і шукані дані. Сюди можуть входити визначення, твердження, формулювання законів тощо. З практичної точки зору "Зв'язок" виражає не стільки спосіб розв'язування, скільки метод перевірки правильності кінцевих результатів.

Приклад математичної моделі:

Задача: Знайти частку двох чисел.



Дано: X, Y - дійсні числа.
Потрібно: знайти Z - частку чисел X і Y .
Зв'язок: $Z = X / Y$
При: $Y \neq 0$

Розв'язування задач:

Побудуйте математичну модель для розв'язування задач:

- 1) обчислити значення функції $y = 1/(x-2)$;
- 2) обчислити площу трапеції, трикутника;
- 3) обчислити модуль числа X не використовуючи стандартну функцію;
- 4) обчислити суму, різницю, добуток, частку двох чисел - X, Y ;
- 5) дано a, b, c . Знайти площу трикутника, дві сторони якого рівні a і b , а кут між ними - c ;
- 6) трикутник заданий координатами $x_1, y_1, x_2, y_2, x_3, y_3$ своїх вершин. Знайти площу і периметр трикутника.

4. **Виконання практичного завдання.** Робота з клавіатурним тренажером. Набір основних команд МП Turbo Pascal.

5. **Домашнє завдання:** Р.3, §1-6, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов.

6. **Підсумок уроку.** Узагальнення навчального матеріалу.



Урок № 25

<i>Тема:</i>	Алгоритми. Властивості алгоритмів. Форми подання алгоритму. Виконавець алгоритму.
<i>Мета:</i>	Розглянути поняття алгоритму, форми подання алгоритму, властивості алгоритмів. Навчити створювати алгоритми у словесно-формульному вигляді. Розвивати логічне та самостійне мислення.
<i>Базові поняття й терміни:</i>	Алгоритм, виконавець.

Хід уроку

1. Мотивація навчальної діяльності. Оголошення теми уроку.

Розв'язування будь-яких задач зводиться до виконання певних дій, команд згідно деякої схеми. В інформатиці таку послідовність команд називають алгоритмом. Які бувають алгоритми? Як вони записуються? В чому відмінність математичних алгоритмів від інформаційних? На ці та інші запитання ми поспробуємо дати відповідь на уроці. Отже, тема уроку: "Алгоритми. Властивості алгоритмів. Форми подання алгоритму. Виконавець алгоритму".

2. Вивчення нового матеріалу.

Пояснення вчителя з ілюстраціями на математичних прикладах.

Алгоритм - чітко задана послідовність дій, спрямованих на досягнення певної мети або розв'язування поставленої задачі.

Приклад:

Вказати послідовність дій, які необхідно виконати для обчислення виразу $(ax+b)/x+c$ при заданих a, b, c, x .

Алгоритм можна описати таким чином:

1. Помножити a на x .
2. До отриманого результату додати b .
3. Отриманий результат поділити на x .
4. До отриманого результату додати c .

Властивості алгоритмів:

1. *Скінченність*. Виконання кожного алгоритму повинно завершуватись за скінчене число кроків.
2. *Результативність*. Виконання алгоритму завжди повинно приводити до певного результату. Воно не може закінчуватись невизначеною ситуацією або ж не закінчуватись взагалі.



3. *Формальність*. Виконавець відповідно до алгоритму повинен отримати результат, не вникаючи в його суть. Ця властивість має особливе значення для автоматизації виконання алгоритмів.
4. *Визначеність*. Будь-який алгоритм повинен бути описаний так, щоб при його виконанні у виконавця не виникло двозначних вказівок. Тобто різні виконавці згідно з алгоритмом повинні діяти однаково та прийти до одного й того ж результату.
5. *Масовість*. За допомогою створеного алгоритму повинен розв'язуватись цілий клас задач.
6. *Зрозумілість*. В алгоритмі повинні бути лише вказівки, які знайомі виконавцеві.

Форми подання алгоритмів:

1. Словесні.
2. Словесно-формульні.
3. Графічні.
4. На одній з мов програмування.

При складанні алгоритмів можна використовувати різні форми подання алгоритмів.

Виконавець алгоритму

Виконавцем алгоритму може бути людина, ПК, система людина-машина, верстат-автомат, робот тощо, яких "навчено" виконувати вказівки алгоритму. Якщо виконавцем є автомат, то вираз "виконавця навчено виконувати вказівку" означає, що автомат може виконати задану вказівку автоматично, без зовнішнього втручання.

Сукупність вказівок, які можуть бути виконані виконавцем, називається системою вказівок виконавця.

4. **Виконання практичного завдання.** Робота з клавіатурним тренажером (набір основних команд МП Turbo Pascal).

5. **Домашнє завдання** Р.4, §1, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов. Опрацювати конспект уроку.

6. **Підсумок уроку.** Узагальнення навчального матеріалу.



Урок № 26

Тема: Базові структури алгоритмів: слідування, розгалуження, повторення. Графічні схеми базових структур алгоритмів.

Мета: Розглянути базові структури алгоритмів. Навчити створювати алгоритми у словесно-формульному вигляді з використанням базових структур.

Базові поняття й терміни: Розвивати логічне та самостійне мислення. Слідування, розгалуження, повторення.

Хід уроку

1. Організаційний етап.

2. Оголошення теми уроку. Мотивація навчальної діяльності (використання міжпредметних зв'язків).

Нагадаємо математику. В ній розв'язування будь-якої задачі зводиться до виконання арифметичних дій, обчислення математичних функцій. Для опису алгоритмів розв'язування задач в інформатиці також використовується певна сукупність команд, вказівок. На відміну від інших наук в програмуванні проведено чіткий поділ всіх команд на групи. На даному уроці нашим завданням буде розгляд базових команд програмування, за допомогою яких можна записати алгоритм розв'язування будь-якої задачі.

3. Вивчення нового матеріалу.

а) Створення проблемної ситуації.

Сформулювати алгоритм розв'язку задач:

1. Додати три числа.

2. Розв'язати квадратне рівняння.

3. Обчислити площу довільного опуклого шестикутника, якщо відомі всі його сторони і діагоналі, проведені до однієї з його вершин. (Формула Герона).

- Що спільного і відмінного в даних алгоритмах?

б) Пояснення нового матеріалу.

Висновок: При конструюванні алгоритмів усі операції можна подати у вигляді комбінацій трьох типів простих операцій, так званих базових алгоритмічних структур:

Слідування. Операція слідування подається у вигляді послідовності двох (або більше) простих операцій, що виконуються одна за одною. Якщо алгоритм складається лише з послідовності простих операцій, то його називають *простим*. Вживається також і інша назва — *лінійний алгоритм*.

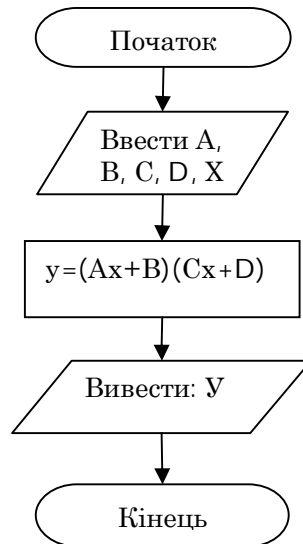


Приклад 1. Скласти алгоритм обчислення і роздрукування значення виразу $y=(Ax+B)(Cx+D)$

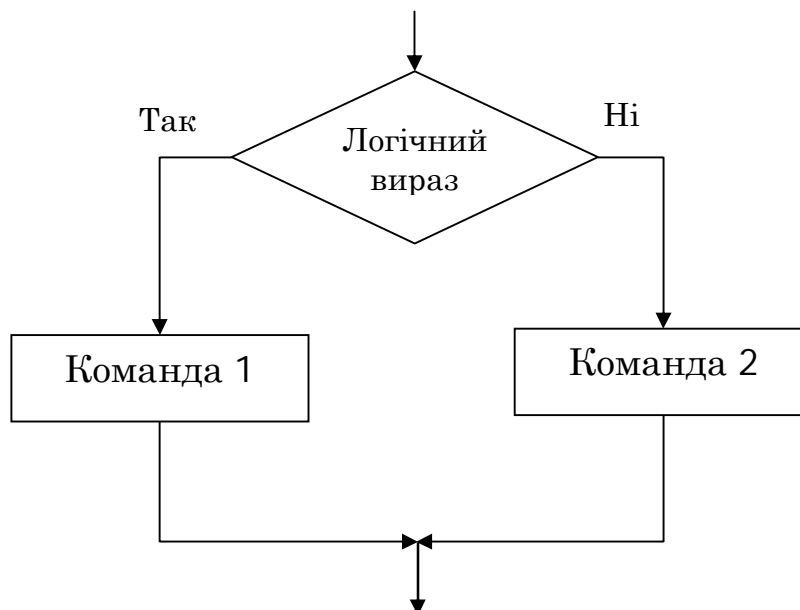
Словесно-формульне представлення алгоритму:

1. Задати значення A, B, C, D, x .
2. Обчислити $y=(Ax+B)(Cx+D)$
3. Надрукувати y .
4. Процес обчислення завершити.

Блок-схема алгоритму



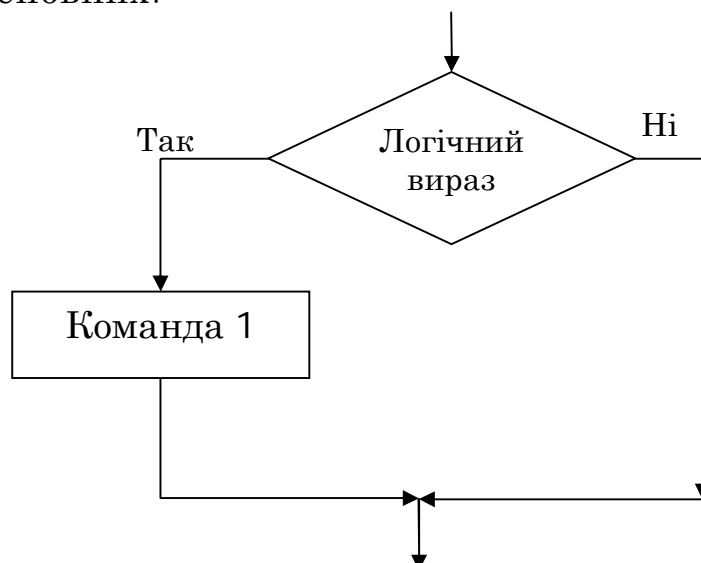
Розгалуження (вибір). Операція розгалуження — це вказівка виконати одну з двох команд: команду 1 або команду 2 залежно від істинності чи хибності деякого твердження (логічного виразу). Якщо твердження правильне, то виконується команда 1 і на цьому виконання операції розгалуження закінчується. Якщо ж твердження хибне, то виконується команда 2 і на цьому виконання операції розгалуження закінчується.



Окремим випадком розгалуження є неповне розгалуження, коли у разі хибного твердження ніякі операції в розгалуженні взагалі не вико-



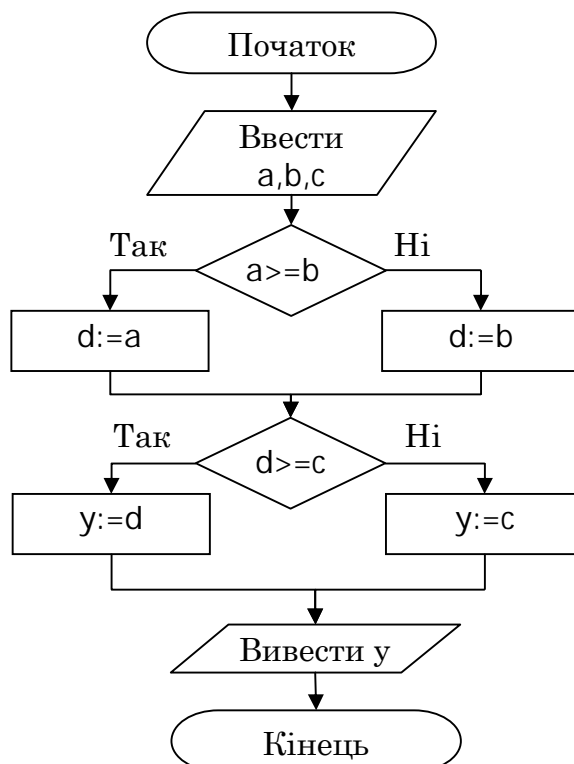
нуються. Повне розгалуження завжди можна подати у вигляді слідування двох неповних.



Твердження може бути утворене з інших тверджень за допомогою логічних операцій НЕ, І, АБО, а замість команди 1 чи 2 може бути декілька команд, які називаються *серією команд*.

Приклад 2. Скласти алгоритм знаходження найбільшого з трьох чисел a , b , c і результат присвоїти змінній y .

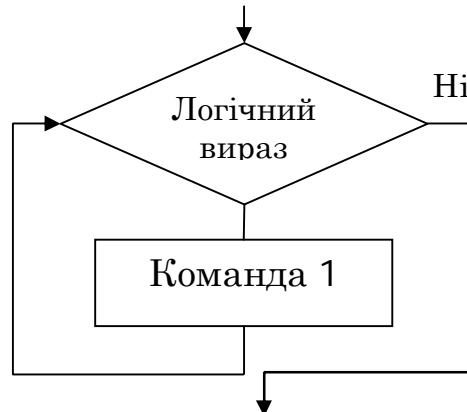
1. Задати значення a , b , c .
2. Якщо $a \geq b$, то $d := a$, інакше $d := b$.
3. Якщо $d \geq c$, то $y := d$, інакше $y := c$.
4. Надрукувати значення y .
5. Процес обчислення завершити.



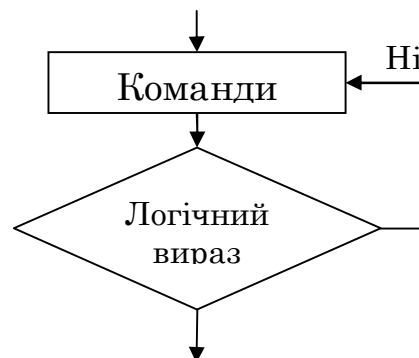


Повторення (цикл). Розрізняють два типи циклів — цикл-ПОКИ і цикл-ДО.

– У структурі цикл-ПОКИ для виконання операції повторення спочатку потрібно визначити, істинне чи хибне твердження (логічний вираз). Якщо твердження істинне, то виконується серія команд 1 і відбувається повернення до визначення істинності твердження. Якщо твердження хибне, то виконання команд вважається закінченим. Отже, поки твердження істинне, відбувається повторне виконання команд і повернення до визначення істинності твердження.



– У структурі цикл-ДО спочатку виконуються команди, а потім визначається істинність твердження. Якщо твердження хибне, то знову виконуються команди і визначається істинність твердження. Якщо твердження істинне, то виконання вказівки вважається закінченим. Отже, виконання команд і визначення істинності твердження повторюються до настання істинності.



У структурі цикл-ПОКИ операція може не виконуватись жодного разу. У структурі цикл-ДО операція буде виконана принаймні один раз.

Кожна з операцій, наприклад, операція перевірки істинності твердження, у свою чергу, може бути подана як сукупність більш простих операцій, об'єднаних у структури розглянутих типів.

Приклад 3: Скласти алгоритм обчислення суми цілих чисел від 1 до 100.

1. Присвоїти початкові значення $S:=0$; $x:=0$.
2. Поки $x \leq 100$, виконувати команди $S:=S+x$; $x:=x+1$.
3. Надрукувати значення S .



4. Процес обчислення значення завершити.

Важливою особливістю розглянутих структур є те, що кожна з них має єдиний вихід і єдиний вхід. При конструюванні алгоритму вихід кожної базової структури приєднується до входу іншої. Таким чином весь алгоритм подається у вигляді лінійної послідовності базових алгоритмічних структур.

3. Виконання практичного завдання. *Інструктаж з техніки безпеки.*

Розв'язування задач:

- скласти алгоритм обчислення площі та периметра прямокутника;
- Обчислити кінетичну $E=mv^2/2$ та потенціальну $P=mgh$ енергії тіла заданої маси m , яке рухається на висоті h зі швидкістю V .
- скласти алгоритм розв'язку квадратного рівняння;
- Ввести два числа. Вивести на екран спочатку менше, а потім — більше.
- Скласти програму, яка перевіряє чи існує трикутник з сторонами А,В,С.
- скласти алгоритм обчислення добутку цілих непарних чисел від 1 до 100.
- Дано натуральне число x . Визначити:
 - 1) кількість цифр у цьому числі;
 - 2) суму цифр у числі;
 - 3) суму першої і останньої цифри числа;
 - 4) чи утворюють цифри числа неспадну послідовність.

4. Домашнє завдання: Р.4, §1, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов.

5. Підсумок уроку. Узагальнення навчального матеріалу. Оцінювання учнів.

**Урок № 27**

Тема: Порядок складання алгоритмів. Поняття програми. Конструювання алгоритмів за методом "зверху донизу".

Мета: Розглянути метод покрокової деталізації при конструюванні алгоритмів.

Ввести поняття структурного програмування, програми.

Продовжувати формувати навички створення алгоритмів з використанням базових структур.

Базові поняття й терміни: Програма, структурне програмування.

Хід уроку**1. Актуалізація опорних знань.****Бесіда з елементами опитування.**

- Назвіть базові структури алгоритмів.
- Базова структура слідування (пояснити, навести приклад).
- Базова структура розгалуження (пояснити, навести приклад).
- Базова структура повторення (пояснити, навести приклад).

2. Мотивація навчальної діяльності (прийом "Практичність теорії").

При розв'язування реальних практичних задач алгоритми та програми є досить складними, їх розмір може сягати тисячі, десятки тисяч стрічок. Як розв'язувати такі задачі? Наприклад програми ОС Windows містять більше одного мільйона стрічок вихідного тексту. Чи можливо в пам'яті втримати такий об'єм інформації? Як в такому випадку поступають програмісти? На ці питання ви знайдете відповідь на даному уроці. Оголошення теми уроку.

Я думаю всім відома фраза "Розділяй і владарюй". Цей принцип використовується і для конструювання складних алгоритмів, в інформації його називають принципом структурного програмування.

3. Вивчення навчального матеріалу.

Структурне програмування — це процес побудови алгоритмів та програм, що виконується в такій послідовності:

1. Попередній аналіз задачі з метою розбиття її на окремі прості частини (модулі). Для цього спочатку складають загальну схему алгоритму, а потім її деталізують.

2. Послідовна (зверху до низу) деталізація частин та складання програм для кожного з модулів. Виділяють основну частину та частини



нижнього рівня. Кожну частину розбивають окремо: спочатку частини верхнього рівня, а потім — нижнього. У кінці частини з'єднують між собою.

Для структурного програмування характерно:

1. Використання трьох базових структур алгоритмів (слідування, розгалуження і циклу) при роботі з кожним модулем. З перерахованими структурами ми ознайомились вище.
2. Коментування текстів програм.
3. Мінімальне використання операторів безумовного переходу, що ускладнюють читання програм.
4. Передбачається система перевірки правильності програми.

4. Виконання практичного завдання. *Інструктаж з техніки безпеки.*

Розв'язування задач:

(Скласти алгоритм. Врахувати, що в ролі виконавця виступає комп'ютер).

1. За двома катетами обчислити площу прямокутного трикутника.

Розв'язок:

- 1). Ввести значення катетів A та B .
 - 2). Обчислити площу за формулою $P=A*B/2$.
 - 3). Надрукувати значення P .
 - 4). Процес обчислення значення завершити.
2. Дано числа x, y . Обчислити їх суму, різницю та добуток.
 3. Дано a, b, c . Знайти площу трикутника, дві сторони якого рівні a і b а кут між ними - α .
 4. Знайти площу сектора, радіус якого рівний R , а відповідний центральний кут - β .
 5. Знайти площу поверхні та об'єм куба, якщо ребро куба рівне a .
 6. Трикутник заданий координатами $x_1, y_1, x_2, y_2, x_3, y_3$ своїх вершин. Знайти площу і периметр трикутника.

4. Домашнє завдання: Р.4, §1, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов.

5. **Підсумок уроку.** Оцінювання роботи учнів на уроці.

**Урок № 28**

Тема: Навчальна алгоритмічна мова. Правила опису алгоритмів навчальною алгоритмічною мовою.

Мета: Навчити учнів описувати алгоритми навчальною алгоритмічною мовою (НАМ). Розглянути синтаксис НАМ та правила опису трьох базових структур алгоритмів. Створювати лінійні алгоритми.

Базові поняття й терміни: Навчальна алгоритмічна мова.

Хід уроку**1. Актуалізація опорних знань. Мотивація навчальної діяльності.****Створення проблемної ситуації.**

Спілкування між споживачами інформації здійснюється за допомогою деякої мови.

- Хто з вас може дати узагальнене означення мови?

- Які типи мов ви знаєте? (Відповіді учнів).

2. Вивчення нового матеріалу.**Пояснення вчителя**

Мова — це сукупність засобів для фіксації повідомлень і передавання їх від джерела інформації до споживача. Розрізняють природні та штучні мови. Штучні мови - мови, які створенні людьми для вирішення власних проблем.

Алгоритмічні мови — мови, призначенні для запису алгоритмів відповідно до певних вимог. Вони відносяться до штучних мов.

Структура опису алгоритму може бути різною залежно від особливостей конкретної алгоритмічної мови. В даному розділі розглядається узагальнений варіант НАМ.

Алфавіт алгоритмічної мови — сукупність символів, які дозволяється використовувати при описанні алгоритмів на тій чи іншій алгоритмічній мові.

Синтаксис алгоритмічної мови — сукупність правил опису алгоритмів алгоритмічною мовою.

Вказівка (команда) — окреме повідомлення про деяку операцію, яку повинен виконати виконавець алгоритму.

Алфавіт навчальної алгоритмічної мови включає в себе великі і малі букви українського і латинського алфавітів, цифри десяткової системи числення, спеціальні символи, символи математичних операцій.



Для більш зрозумілого і виразного запису алгоритмів алфавіт алгоритмічної мови доповнено службовими словами.

Правила опису алгоритмів:

У першому рядку записується слово **програма** після якого записується назва алгоритму.

У другому рядку записується слово **початок** або **поч.**, після якого записуються дії, що складають алгоритм.

Останнім рядком опису алгоритму має бути слово **кінець** або **кін.** у тій самій позиції що й початок.

Алгоритм, описаний НАМ, має вигляд:

```
Заголовок алгоритму
початок
    серія вказівок
кінець
```

де "серія вказівок" — послідовність вказівок, що виконуються в тому ж порядку, в якому вони записані.

Заголовок алгоритму — містить в собі ім'я алгоритму, перелік його аргументів і результатів з зазначенням їх величин.

Заголовок алгоритму має вигляд:

```
програма ім'я _алгоритму;
```

```
    константи – список констант;
```

```
    змінні – список змінних величин та їх тип;
```

Виділенні слова називають службовими і використовують при записі довільного алгоритму.

Тіло алгоритму — частина алгоритму, яка знаходиться між словами початок і кінець.

Структура алгоритму, описана навчальною алгоритмічною мовою, в цілому має вигляд:

```
програма ім'я _алгоритму;
```

```
    константи – список констант;
```

```
    змінні – список змінних величин та їх тип;
```

```
початок
```

```
    серія команд;
```

```
кінець.
```

Основні вказівки навчальної алгоритмічної мови.

Будь-який алгоритм можна описати з використанням вказівок лише трьох типів:

- 1) вказівок про безумовне виконання деякої операції;
- 2) вказівок розгалуження;
- 3) вказівок про повторне виконання деякої операції.

До вказівок першого типу належать вказівки присвоєння $Y:=A$ (де A — вираз, Y — ім'я результату, $:=$ — знак присвоєння).



Вказівки другого типу записуються у вигляді:

ЯКЩО твердження

ТО

серія 1

ІНАКШЕ

— повне розгалуження

серія 2

ЯКЩО твердження

ТО

серія 1

— неповне розгалуження

Якщо "серія" команд містить більше однієї вказівки то вони обмежуються словами **початок**, **кінець**.

Узагальненням вказівки розгалуження є вказівка вибір, за допомогою якої описується розгалуження в трьох і більше напрямках:

випадок с із

твердження 1: серія 1;

твердження 2: серія 2;

..... ;

твердження k: серія k;

кінець;

Вказівки третього типу описуються у вигляді:

поки твердження істинне виконати

початок

серія;

— цикл-поки

кінець;

повторювати

серія ;

— цикл-до

покине твердження;

Використовується також вказівка повторення з параметром

для x := a до b виконати

початок

серія;

кінець;

де A, B— вирази, що задають відповідно початкове і кінцеве значення параметру циклу X.

3. Виконання практичного завдання. ІТБ.

Скласти алгоритм:

а) периметр квадрата, площа якого рівна a

б) площу рівностороннього трикутника, периметр якого рівний P.

в) робота сили тяжіння при переміщенні тіла масою m на відстань S.

г) об'єм куба з ребром - a.

4. **Домашнє завдання:** Р.4, §1, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов. Підготувати повідомлення на тему: "Мови програмування", "Історія створення мови програмування TurboPascal "

5. **Підсумок уроку.** Узагальнення навчального матеріалу.



Урок № 29

Тема: Поняття програми. Мова програмування. Переклад програм з однієї мови на іншу. Поняття про транслятори, систему програмування. Поняття про середовище програмування TurboPascal.

Мета: Розглянути поняття: програма, мова програмування, інтерпретатор, компілятор. Навчити працювати в середовищі програмування TurboPascal, запускати програми на виконання.

Базові поняття й терміни: Мова програмування.

Хід уроку

1. Організаційний момент.
2. Оголошення теми, мети уроку.
3. Вивчення нового матеріалу.

Повідомлення учнів. Очікувані відповіді:

Мова програмування Паскаль є однією з найбільш поширених мов програмування. Була створена в 1968-1971 рр. швейцарським професором Ніклаусом Віртом, як засіб для навчання програмування. Завдяки принципам структурного програмування і покрокової деталізації, програми мовою Паскаль більш зрозумілі ніж програми іншими мовами.

Мова Паскаль є структурованою мовою програмування, оскільки використовує тільки 4 основні структури:

- 1) вказівки безумовного виконання;
- 2) вказівки розгалуження;
- 3) вказівки повторення;
- 4) виділення допоміжних програм.

Головною відмінністю мови від інших неструктурних мов є деякі обмеження використання вказівок безумовного переходу.

Весь матеріал з мови Turbo Pascal викладений стосовно версій Turbo Pascal 6.0-7.0. В новій версії Turbo Pascal 7.0 є деякі доповнення до попередніх версій що зрівняло її по можливостях з мовою C++.

Пояснення вчителя.

Програма - впорядкована послідовність дій для ПК, використання якої реалізує алгоритм розв'язку будь-якої задачі.

Транслятор - програма, яка перекладає програму з мови програмування в машинний код.



Переведення програм в машинні коди та їх виконання може здійснюватися шляхом інтерпретації або компіляції.

Під час *інтерпретації* кожна вказівка програми аналізується на предмет виявлення в ній синтаксичних і логічних помилок, переводиться в машинні коди, а потім виконується. Тобто виконання програми проходить шляхом їх покомандної інтерпретації.

Під час *компіляції* в усьому тексті виявляються синтаксичні та логічні помилки, після цього вся програма переводиться в машинний код і лише тоді виконується.

Система програмування - частина базового програмного забезпечення, яка підтримує процес програмування на ПК.

Система програмування містить:

- інтерпретатор (для виконання програм);
- редактор (для складання і введення програм);
- транслятор або компілятор (для перекладу програм з вхідної мови на машинну).

Система програмування Turbo Pascal об'єднує в собі:

- текстовий редактор;
- компілятор;
- компоновник;
- відлагоджувач;
- систему підказки.

Система програмування TurboPascal складається з великої кількості файлів. Головні з них:

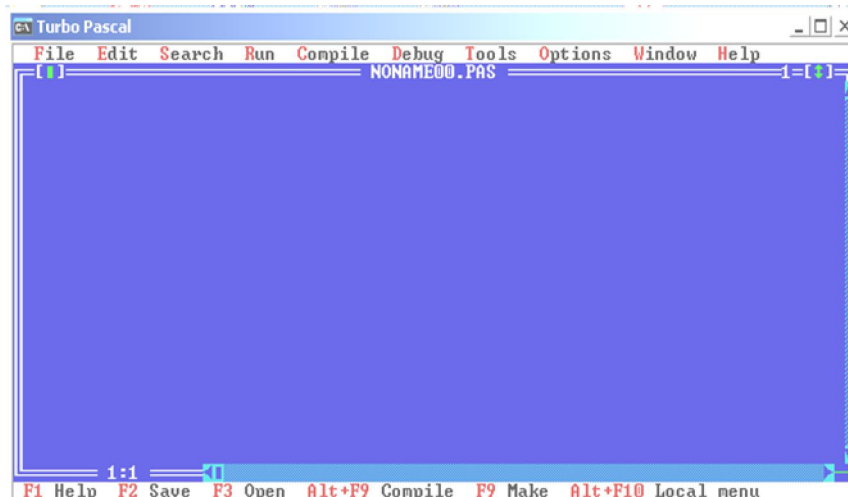
- 1) Turbo.exe - забезпечує роботу в середовищі програмування;
- 2) Turbo.tpl- містить бібліотеку стандартних підпрограм;
- 3) Turbo.hlp-забезпечує введення на екран підказки.

Поняття про середовище програмування TurboPascal.

Щоб ввійти в середовище TurboPascal, потрібно:

1. Ввійти в каталог системи програмування.
2. Запустити на виконання командний файл turbo.exe.

На екрані з'явиться головне меню системи.





У верхньому рядку подано команди головного меню, а в нижньому - рядок статусу, в якому перелічені імена функціональних клавіш, призначених для виконання деяких операцій в конкретній ситуації.

Щоб перейти в головне меню, потрібно натиснути клавішу *F10*. Щоб вийти з головного меню і повернутись в редактор тексту - *Esc*.

Виконати команду означає: з допомогою клавіш управління курсором виставити курсор на потрібну команду і натиснути клавішу *Enter*. (Виконати команду можна за допомогою "миші" або комбінацією *Alt + буква*, яка виділена в назві команди).

Кожна команда головного меню має одне або декілька підменю.

Типовий порядок створення нового файлу:

- ввійти в головне меню;
- виконати команду *File*;
- вибрати і виконати команду *New*. На екрані з'явиться порожнє вікно редактора з умовною назвою *Noname.pas*;
- набирати текст програми. В кінці кожного рядка натиснути *Enter*;
- виконати програму. Натиснути *Ctrl + F9*. Або ввійти в головне меню і вибрати і виконати команду *Run*;
- виправити при необхідності помилки;
- знову виконати програму;
- переглянути результати, натиснувши *Alt + F5*;
- записати програму в файл на диск, ввійти в головне меню виконати команду *File*;
- Виконати команду *Save*. Ввести ім'я і натиснути *Enter*.

Алфавіт. Величини та їх опис. Змінні та константи. Ідентифікатори величин.

Алфавіт TurboPascal включає в себе:

- 1) 26 великих і малих букв латинського алфавіту, знак підкреслення;
- 2) цифри 0...9;
- 3) спеціальні символи:

+	<	@	`
-	>	.	#
*	[]	,	\$
/	()	:	^
=	{ }	;	

Проміжок - незаповнена позиція в тексті, займає одну позицію.

4. Комбінації спеціальних символів утворюють складені символи:

:= - присвоєння	(.) - альтернатива { }
<> - не дорівнює	<= - менше або дорівнює
.. - діапазон значень	>= - більше або рівне

5. Службові слова-зарезервовані.



Службові слова мають раз і назавжди визначену форму запису і зміст.

Ідентифікатори

Для позначення констант, змінних, типів, процедур і функцій, файлів і програм використовують *імена (ідентифікатори)*.

Правила запису ідентифікаторів:

1. Ідентифікатори починаються тільки з букви або знаку підкреслення.
2. Ідентифікатор може складатися з букв, цифр, і знаку підкреслення.
3. Між двома ідентифікаторами повинен бути хоча б один проміжок.
4. Максимальна довжина ідентифікатора 127 символів, але ПК розрізняє тільки 63.
5. Ідентифікатор не повинен співпадати з службовими словами.
6. Великі і малі букви не розрізняються.

Константи і змінні

Константи - величини, значення яких встановлені в описовій частині програми і в процесі виконання програми не змінюються.

Для оголошення констант в програмі використовується службове слово - **CONST**.

Формат:

CONST < Ідентифікатор> = < значення константи>

Наприклад:

```
CONST a = 13.5;  
      f1 = 15;
```

Зарезервовані константи

Ідентифікатор	Тип	Значення	Опис
TRUE	boolean	true	"Істина"
FALSE	boolean	False	"Хибно"
MAXINT	integer	32767	Макс. ціле

Константам в програмі не можна присвоювати нові значення після того як вони були описані.

Змінні - величини значення яких змінюється в процесі виконання програм.

Для опису змінних використовується службове слово **VAR**.

Формат:

VAR <список ідентифікаторів>: < тип>

Наприклад:

```
VAR i,x,r1,d : real;  
    t,ca : integer;
```




3. Виконання практичного завдання. Інструктаж з техніки безпеки.

1. Наберіть, відредагуйте та запишіть на диск (ім'я – Z1.pas) текст програми: (3 бали)

```
program z_1;  
  var x,y,c : real;  
      d : real;  
begin  
  readln(x,y,c);  
  d:=x*y*c;  
  writeln('d=',d:8:2)  
end.
```

2. Запустіть програму на виконання, введіть довільні значення та запишіть отриманий результат. (2 бали)
3. Створіть нове вікно. (1 бал)
4. Наберіть, відредагуйте, запишіть на диск (ім'я – Z2.pas) та запустіть на виконання програму: (3 бали)

```
program z_2;  
  var a,b,c : real;  
      p,s : real;  
begin  
  readln(a,b,c);  
  p:=(a+b+c)/2;  
  s:=sqrt(p*(p-a)*(p-b)*(p-c));  
  writeln('s=',s:5:3);  
end.
```

Результат виконання запишіть в зошит.

5. Відкомпілюйте програму на диск та запустіть її на виконання. Результат запишіть в зошит. Для чого призначена програма? (3 бали)

4. **Домашнє завдання:** Р.4, §1, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов.

5. **Підсумок уроку.** Узагальнення навчального матеріалу. Оцінювання учнів.

**Урок № 30**

Тема: Величини. Основні характеристики величин. Вказівка присвоювання.

Мета: Ввести поняття величин та їх головних характеристик.

Розглянути принципи роботи вказівок введення, виведення інформації, вказівки присвоєння. Удосконалювати вміння створювати алгоритми мовою програмування.

Розвивати вміння аналізувати умову задачі, вибирати оптимальний метод розв'язання.

Базові поняття й терміни: Типи величин.

Хід уроку**1. Актуалізація опорних знань.****Усне опитування.**

- Запишіть загальну структуру програми;
- В чому різниця між змінними та константами;
- Назвіть правила опису алгоритмів НАМ.

2. Вивчення нового матеріалу.**Пояснення вчителя.****Величини**

Повідомлення N разом з відповідною йому інформацією I називають величиною. Інформацію I називають значенням величини, а повідомлення N — позначенням величини. При цьому говорять, що позначенню N відповідає значення I .

Величина — об'єкт, який має ім'я і значення конкретного типу.

Операнд — об'єкт алгоритму, над яким виконується деяка вказівка.

Тип - множина значень однакової природи разом із набором операцій, які над ними виконуються.

Кожна змінна повина бути описана тільки один раз на початку програми після слова **VAR**

Скалярні типи даних

Цілі або цілочисельні типи

Тип	Діапазон
byte	0...255
shortint	-128...127
integer	-32768...32767
word	0...65535
longint	-2147483648...2147483647



При виході значень даних цілого типу за вказаний діапазон помилки виконання програми не буде але результат буде не правильний.

Дійсні типи

Тип	Діапазон
real	$2.9 \cdot 10E-39 \dots 1.7 \cdot 10E38$
singl	$1.5 \cdot 10E-45 \dots 3.4 \cdot 10E38$
double	$5.0 \cdot 10E-324 \dots 1.7 \cdot 10E308$
extended	$1.9 \cdot 10E-4951 \dots 1.1 \cdot 10E4932$
comp	$-2E+63+1 \dots 2E+63-1$

Літерний тип

Тип	Діапазон
char	кодова таблиця ПЕОМ

Булевий тип

Тип	Діапазон
boolean	True , False

Структуровані типи даних

Рядковий тип.

Рядок - послідовність символів кодової таблиці ПК.

Для опису даного типу використовується ідентифікатор String, після якого в квадратних дужках записується значення максимальної довжини рядка. Якщо значення не вказується, то довжина рядка дорівнює 255 байт.

Наприклад:

```
Var a,c :String[25];
    v : String;
```

Інші типи даних.

В Паскаль використовуються такі типи даних: масиви, множини, записи, файли, процедурні типи, об'єкти. Всі вони потребують окремого детального вивчення.

Вказівки присвоювання, введення й виведення.

Вказівка присвоювання.

Вказівка присвоювання має вигляд:

```
<ім'я змінної>:=<вираз>;
```

Змінна і вираз повинні бути одного типу. Виняток становить випадок дійсної змінної, коли вираз може бути і цілого типу.

Виконується вказівка присвоєння таким чином: спочатку обчислюється значення виразу в правій частині вказівки присвоювання і це значення присвоюється змінній в лівій частині (заноситься в пам'ять ПК).

Вказівка введення даних

Виведення даних здійснюється вказівками **Read** і **Readln**.

Загальний вигляд:

```
Read(список);
Readln(список);
```



В списку перелічуються через кому імена змінних, значення яких вводяться з клавіатури.

Після введення значення змінної натискається **Enter**. Потім значення другої і знову **Enter** і т.д. Якщо потрібно ввести значення декількох змінних, то можна записати їх значення в одному рядку, відділивши їх проміжками і натиснути **Enter**.

Між вказівками **Read** / **Readln** є різниця.

Після виконання **Read** курсор залишиться в цьому ж рядку. Після виконання **Readln** буде зроблено перехід у наступний рядок (при введенні даних у файл).

Наприклад:

```
Read(a,b,c);  
Readln(f,g);
```

Вказівка виведення даних

Виведення даних виконується вказівками **Write** і **Writeln**.

Загальний вигляд цих вказівок:

```
Write(список);  
Writeln(список);
```

У списку перелічується через кому імена змінних, вирази, підказки і т.д. Якщо у списку є вираз, то він попередньо обчислюється, а вже потім результат виводиться на екран.

Між вказівками **Write** і **Writeln** є різниця. При виконанні вказівки **Write** значення змінних виводиться в один рядок, так як вони записані.

При виконанні вказівки **Writeln** значення кожної змінної виводиться в окремому рядку, тобто після виведення значення змінної X буде зроблено перехід у наступний рядок і т.д.

Для зручності введення даних користуються поєднанням вказівок **Read** і **Write**.

Друкування результатів за допомогою принтера в цілому співпадає з виведенням даних на екран, тільки у вказівках **Write** або **Writeln** потрібно записати ім'я файлу **Lst** і в розділі **Uses** вказати ім'я модуля **Printer**.

Існує ще один спосіб керуванням виведення інформації. Можна вказати ширину поля для кожного елемента виведення, форму запису числових значень. Здійснюється це з допомогою одного або двох цілих чисел, які записуються в вказівці **Write** після відповідного елемента виведення і відділяється від нього і між собою двокрапкою- ":".

Наприклад:

```
Write(f,a);  
Writeln(b:8:5); - дійсний тип  
Writeln(b:5); - цілий тип  
Writeln('y=',y:4:2); - дійсний тип з підказкою
```



Складена вказівка - це послідовність довільних вказівок програми, відокремлених одна від одної крапкою з комою, взятих в операторні дужки - службові слова **begin .. end**.

Після службового слова **end** ставиться крапка з комою. У деяких випадках, коли складена вказівка використовується у вказівках розгалуження, крапка з комою може не записуватись.

Не існує ніяких обмежень на характер вказівок, які входять в складену вказівку. Серед них можуть бути і інші складені вказівки.

Складена вказівка сприймається як одне ціле і може записуватись у довільному місці програми, де синтаксис мови дозволяє наявність вказівки.

3. Практичне завдання. Інструктаж з техніки безпеки.

Скласти програму.

I варіант

I рівень

- обчислити суму двох чисел X та Y .

II рівень

- обчислити периметр та площу прямокутника з сторонами A та B .

III рівень

- квадрат заданий довжиною сторони. Обчислити довжину та площу вписаного кола.

IV рівень

- Підприємство поклало в банк на депозитний рахунок S гривень під 2% місячних. Яку суму зніме підприємство через півроку. По вкладу нараховуються складні відсотки.

II варіант

I рівень

- обчислити добуток двох чисел X та Y .

II рівень

- обчислити периметр та площу квадрата з стороною A .

III рівень

- квадрат заданий довжиною сторони. Обчислити довжину та площу описаного кола.

IV рівень

- Підприємство взяло в банку кредит на S гривень під 3% місячних на півроку. Яку суму повинно буде виплатити підприємство? По кредиту нараховуються складні відсотки.

4. Домашнє завдання: Р.5. Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов.

5. Підсумок уроку. Оцінювання учнів.

**Урок № 31**

Тема: Поняття основного та допоміжного алгоритму. Процедури та функції. Формальні та фактичні параметри. Глобальні та локальні змінні. Розв'язування задач.

Мета: Ввести поняття основного та допоміжного алгоритму.

Показати різницю між процедурами та функціями, формальними та фактичними параметрами, глобальними та локальними змінними.

Навчити розв'язувати задачі з використання підпрограм.

Базові поняття й терміни: Підпрограми, процедури, функції

Хід уроку

1. Організація класу.
2. Оголошення теми, мети уроку.
3. Вивчення нового матеріалу.

Пояснення нового матеріалу

Підпрограма - це логічно завершена частина програми, яка має ім'я і може бути викликана з будь-якого місця основної програми.

В мові програмування Турбо Паскаль підпрограми реалізуються за допомогою процедур і функцій.

Процедура (функція) - це незалежна іменована частина програми, яку можна викликати по імені для виконання наперед визначених дій. Всі процедури і функції поділяють на дві групи: вбудовані і визначені користувачем. Перші входять до складу мови програмування і викликаються для виконання по фіксованому імені (математичні функції, графічні процедури та ін.). другі розробляються і іменуються користувачем.

Процедури і функції повинні бути описані в розділі опису даних. Середині кожної процедури (функції) можуть використовуватись інші процедури (функції) але описані вище її по тексту.

Структура процедури (функції) повторює структуру програми, тільки замість слова *Program* записується слово *Procedure* або *Function*.

Procedure ім'я(список формальних параметрів);

розділ опису даних;

Begin

тіло процедури;

End;

Function ім'я(список формальних параметрів):тип ;

розділ опису даних;



```

Begin
  тіло функції;
End;
```

Відмінність між процедурою і функцією полягає в тому, що функція повертає в точку свого виклику одне значення тип якого повинен співпадати з типом функції. В тілі функції повинен бути використаний оператор присвоювання:

```
ім'я функції := вираз;
```

В списку формальних параметрів перераховуються імена параметрів (а не конкретні значення) і вказується їх тип.

Наприклад:

```

Procedure Geron(a,b,c: real);
Function factorial(n:integer):integer;
Procedure P2(a,d:real,c:integer,f:char);
```

Після опису процедур(функцій) їх можна викликати замінивши формальні параметри фактичними (конкретними значеннями).

Наприклад:

```

Procedure Geron(3.5,4.2,6);
Function factorial(10);
Procedure P2(-4,2.3,11,'X');
```

- Розглянемо використання підпрограм на такому прикладі:

З трьох заданих чисел знайти максимальне і добути з нього корінь кубічний, потім знайти мінімальне і піднести його до кубу.

- Запишемо словесний запис алгоритму:

Початок

ввід даних;

пошук максимального і мінімального числа;

обчислення і вивід $\max^{1/3}$;

обчислення і вивід \min^3 ;

Кінець.

- Переведемо даний алгоритм на мову програмування:

```

Program Z7_2;
  var x1,y1,z1,min1,max1:real;
  Procedure M(x,y,z:real;var min,max:real);
    { процедура пошуку мінімального і максимального числа }
begin
  max:=x;
  if max<y then max:=y;
  if max<z then max:=z;
  min:=x;
  if min>y then min:=y;
  if min>z then min:=z;
end;
```



```

Function M1(max2:real):real;
begin
  M1:=exp(1/3*ln(max2));
end;
Function M2(min2:real):real;
begin
  M2:=exp(3*ln(min2));
end;
BEGIN
  read(x1,y1,z1);
  M(x1,y1,z1,min1,max1);
  M1(min1);
  M2(max1);
  writeln(min1,'  'max1);
  writeln(M1,'  'M2);
END.

```

3. Практичне завдання. Інструктаж з техніки безпеки.

Розв'язування задач.

В середовищі програмування TurboPascal скласти програми для розв'язку задач:

Завдання до 9 балів

I варіант

а) Обчислити значення виразу: $y = \frac{\sqrt{\operatorname{tg}x + 2}}{\operatorname{tg}^2x - 1}$; створити функцію Tan для

обчислення функції tg.

б) Обчислити значення виразу: $y = \frac{\sqrt{\operatorname{tg}x + 2}}{\operatorname{tg}^2x - 1}$; створити процедуру Tan для

обчислення функції tg.

II варіант

а) Обчислити значення виразу $y = \frac{\operatorname{ctg}^2x + 2}{x + \frac{1}{\operatorname{ctgx} + 2}}$; створити функцію Ctg для

обчислення функції ctg.

б) Обчислити значення виразу $y = \frac{\operatorname{ctg}^2x + 2}{x + \frac{1}{\operatorname{ctgx} + 2}}$; створити процедуру Ctg для

обчислення функції ctg.

III варіант

а) Обчислити значення виразу $y = \frac{2z - 4x}{z + 2} - z^2$, де $z = x^2 - 4x$.

Створити функцію t1 для обчислення значення – z.

б) Обчислити значення виразу $y = \frac{2z - 4x}{z + 2} - z^2$, де $z = x^2 - 4x$.

Створити процедуру t1 для обчислення значення – z.

**Завдання до 12 балів****10 балів**

1. Створити підпрограму, яка б за даними дійсними координатами двох точок визначала довжину відповідного відрізка та довжину його проекції на вісь ОХ. У програмі організувати виклик цієї підпрограми.
-

11 балів

1. Дано дві пари дійсних чисел, що є координатами відповідно лівого верхнього та правого нижнього кутів прямокутника, сторони якого паралельні осям координат. Створити підпрограму, що визначатиме довжини сторін цього прямокутника.
-

12 балів

1. Дано дійсні координати чотирьох точок на площині. Визначити, які трійки з них утворюють трикутники і обчислити площу більшого трикутника, створивши для цього функції обчислення довжини відрізка та площі трикутника.
-

4. **Домашнє завдання:** Р.4, §1, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов.

5. **Підсумок уроку.** Узагальнення навчального матеріалу. Оцінювання учнівських робіт.



Урок № 32

<i>Тема:</i>	Вказівка розгалуження та вибору, їх опис навчальною алгоритмічною мовою та мовою програмування. Розв'язування задач з використанням вказівки розгалуження.
<i>Мета:</i>	Розглянути вказівки розгалуження, безумовного переходу та вибору, їх опис навчальною алгоритмічною мовою та мовою програмування. Навчити розв'язувати задачі з використанням даних вказівок. Розвивати самостійне, логічне мислення.
<i>Базові поняття й терміни:</i>	Вказівка розгалуження, вказівка вибору.

Хід уроку

1. Організація класу.

2. Актуалізація опорних знань.

- Назвіть базові структури алгоритмів.
- Опишіть базові структури алгоритмів алгоритмічною мовою.
- Запишіть основні позначення блок-схем.

3. Оголошення теми, мети уроку.

4. Вивчення нового матеріалу.

а) Мотивація навчальної діяльності.

Якщо при розв'язуванні задачі виникає ситуація, коли при виконанні деякої умови необхідно виконати певні дії, а в випадку невиконання умови інші, то використовується вказівка розгалуження.

б) Пояснення нового матеріалу.

Вказівка розгалуження записується у вигляді:

IF логічний вираз THEN вказівка1(и) ELSE вказівка2(и);

Читається так: *Якщо* значення логічного виразу істинне *То* виконується вказівка1(и) *Інакше* виконується вказівка2(и).

Логічним виразом називають вираз, внаслідок якого отримується логічне значення *True* (істина) або *False*.

Існує короткий запис вказівки розгалуження (неповне розгалуження):

IF логічний вираз THEN вказівка(и);

Відмінність полягає в тому, що при набутті логічним виразом значення *False* виконуються вказівки, які стоять після вказівки розгалуження.

Якщо після слів **Then** або **Else** стоїть декілька вказівок то вони обмежуються операторними дужками - **Begin** і **End**.



Розглянемо використання вказівки розгалуження на прикладі класичної задачі:

З трьох заданих чисел визначити максимальне.

```

Program Maximum;
  var a,b,c:real;           {задані числа}
      max:real;           {допоміжна змінна}
Begin
writeln('Введіть три числа');
readln(a,b,c);           {введення чисел}
if a>b                   {визначення більшого з двох}
  then max :=a
  else max:=b;
if max<c                 {визначення max}
  then max:=c;
writeln('max=',max)     {вивід max на екран}
End.

```

Алгоритм даної задачі можна побудувати без використання допоміжної змінної.

Примітка Якщо логічний вираз складається з декількох логічних виразів то такий вираз називають складеним, а логічні вирази беруться в дужки, які об'єднують за допомогою службових слів **And** (і), **Or**(чи), **Not**(заперечення). Навести таблиці їх значень.

Наприклад:

```
IF (a=b) and (a+c>r) THEN ....
```

Вказівка безумовного переходу (*Goto*).

Вказівки в програмі виконуються, як правило в порядку її запису. Змінити цей порядок можна за допомогою вказівки переходу, яка записується у вигляді:

Goto мітка;

Вказівка безумовного переходу може бути записана в будь-якому місці програми. Вона дозволяє перейти до мітки передавши управління вказівці, що знаходиться за міткою. Мітки повинні бути описані в розділі *Label*.

Наприклад:

```

K1 : c:=c+2;
    if c<10 then Goto K1;

```

Вказівка вибору (Case).

Вказівка розгалуження використовується, коли в залежності від умови розв'язок розгалужується на дві гілки. І хоча дану вказівку можна використовувати для розгалуження на більше гілок, вигідніше використати в такому випадку вказівку вибору.



Вказівка вибору записується у вигляді:

CASE вираз OF

мітка1 : вказівка1;

мітка2 : вказівка2;

.....

міткаN : вказівкаN

ELSE вказівка

END;

Вираз який знаходиться між словами **Case** і **Of** повинен належати до одного з порядкових типів даних (всі типи, крім дійсного або рядкового).

Мітки варіантів - константи, тип яких повинен співпадати з типом виразу. Якщо одній вказівці відповідає декілька міток то їх перераховують через кому.

Виконання вказівки вибору відбувається так: спочатку обраховується значення виразу, а тоді виконується вказівка мітка якої співпала з отриманим значенням. Якщо такої мітки нема то виконується вказівка, яка стоїть після слова **Else**.

Розглянемо роботу вказівки вибору на прикладі програми:

Скласти програму, яка визначає квартал року за порядковим номером місяця.

Program KVARTAL;

Var M: integer;

Begin

 readln(M);

Case M div 3 **of**

 0 : write('1 квартал');

 1 : write('2 квартал');

 2 : write('3 квартал');

 3 : write('4 квартал');

end

End.

3. Практичне завдання. Інструктаж з техніки безпеки.

Розв'язування задач.

В середовищі програмування TurboPascal скласти програми для розв'язку задач:

Обчислити значення функції згідно свого варіанту:

№	Функція	Умова
1	$Y = \begin{cases} at^2 \ln(t) \\ 1 \\ e^{at} \cos(bt) \end{cases}$	$1 \leq t \leq 2$ $t < 1$ $t > 2$



2	$Y = \left\{ \begin{array}{l} x^2 - 7/x^2 \\ ax + 7 \cdot x \\ \lg(x + 9) \end{array} \right.$	$X < 1,3$ $X = 1,3$ $X > 1,3$
3	$Y = \left\{ \begin{array}{l} ax^2 + bx + c \\ a/x + 7 \cdot (x^2 + 1) \\ (a + bx) / x + 1 \end{array} \right.$	$X < 1,2$ $X = 1,2$ $X > 1,2$
4	$Y = \left\{ \begin{array}{l} x^2 + 6 / (x + 1) \\ bx^2 - 8x \\ \lg(x + 2b) \end{array} \right.$	$X < 1,4$ $X = 1,4$ $X > 1,4$
5	$Y = \left\{ \begin{array}{l} 1.5 \cos^2 x + 5a \\ 1.8 ax + \operatorname{tg} a \\ (x - 2)^2 + 6 \end{array} \right.$	$X < 2$ $X = 2$ $X > 2$
6	$Y = \left\{ \begin{array}{l} x \cdot (x - a) \\ x \operatorname{Sin}(ax) \\ e^{-ax} \operatorname{Cos}(ax) \end{array} \right.$	$X < a$ $X = a$ $X > a$
7	$Y = \left\{ \begin{array}{l} bx - \lg(bx) \\ \\ bx + \lg(x) \end{array} \right.$	$X < 1$ $X = 1$ $X > 1$
8	$Y = \left\{ \begin{array}{l} (5x^2 + 1) / x + t \\ x + t + 1 / x \\ \operatorname{Cos}(x) + tx^2 \end{array} \right.$	$X < 0,5$ $X = 0,5$ $X > 0,5$

4. **Домашнє завдання:** Р.5, §2.3, 2.5, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов.

5. **Підсумок уроку.** Узагальнення навчального матеріалу.

**Урок № 33**

Тема: Вказівка повторення та її опис навчальною алгоритмічною мовою і мовою програмування. Розв'язування задач з використанням вказівки повторення.

Мета: Розглянути типи вказівок повторення та їх запис алгоритмічною мовою та мовою програмування.

Навчити використовувати вказівки повторення для розв'язування прикладних задач.

Базові поняття й терміни: Вказівки повторення, цикл.

Хід уроку

1. **Організація класу.**
2. **Актуалізація опорних знань.**
Запишіть загальний вигляд вказівки:
 - розгалуження;
 - вибору;
 - безумовного переходу.
3. **Оголошення теми, мети уроку.**
4. **Вивчення нового матеріалу.**

Пояснення вчителя.

При створенні програм часто виникає необхідність повторити деяку команду або групу команд багато раз. Для цього існують спеціальні конструкції - цикли, які вказують скільки разів потрібно повторити команду(групу команд), або до яких пір повторяти дані команди. В TurboPascal існує три види циклів, які починаються службовими словами **While** (поки), **Repeat** (повторяти), **For** (для).

1.1 Вказівка повторення з передумовою (while).

Вказівка повторення з передумовою записується у вигляді:

WHILE логічний вираз **DO** вказівка;

або

```

WHILE логічний вираз DO
  BEGIN
    вказівка ;
    .....;
    вказівка ;
  END;

```



Читається так: *Поки* логічний вираз істинний *Виконувати* вказівку(и). Виконувану частину часто називають тілом циклу. Працює дана конструкція наступним чином. Спочатку обчислюється значення логічного виразу. Якщо воно рівне *False* (хиба) то тіло циклу не виконується ні разу. Якщо при початковій перевірці значення логічного виразу *True* (істина) то виконується тіло циклу після службового слова **Do**, після чого здійснюється повернення на початок циклу і все повторюється спочатку. В тілі циклу необхідно передбачити зміну даних для логічного виразу, інакше його значення може постійно бути *True*, що призведе до зациклювання (цикл виконується безкінечно довго).

Розглянемо приклад простої програми.

1. Обчислити суму парних чисел на проміжку від 0 до 100.

```

Program SUMMA;
  var i,s: integer;
Begin
  s:=0;
  i:=2;           {початкові значення }
While i<= 100 do
  begin
    s:=s+i;
    i:=i+2;     {зміна даних для лог. виразу}
  end;
write('S=',s)
End.

```

1.2 Вказівка повторення з післяумовою (**Repeat**).

Вказівка повторення з післяумовою записується у вигляді:

```

REPEAT
  вказівка (и);
UNTIL логічний вираз;

```

Читається так: *Повторювати* вказівку(и) *До тих пір*, *Поки* логічний вираз не стане істинним (*True*). Тобто даний цикл завершується, коли логічний вираз приймає значення *True*, а не *False*, на відміну від циклу **while**. Так як умова виходу з циклу перевіряється в кінці, то тіло циклу виконується завжди, як мінімум один раз. Якщо логічний вираз завжди буде приймати значення *False* то відбудеться зациклювання.

Цикл **Repeat** використовується в ситуаціях, коли виконується деяка група вказівок, а потім, в залежності від результату, або продовжується робота програми далі, або дана група вказівок виконується знову. Класичним прикладом є обчислення виразу з заданою точністю (точність досягнута - закінчуємо обчислення, ні - продовжуємо).

Розглянемо роботу даного циклу на прикладі програми.



Поділити задане трьохзначне число на 3 з точністю до цілих не використовуючи дії ділення.

```
Program DIL;  
  Var x : real;  
      i : integer;  
Begin  
  readln(x);  
  i:=0;  
  Repeat  
    x:=x-3;  
    i:=i+1  
  until x<3;  
  write (i)  
End.
```

1.3 Вказівка повторення (FOR - TO - DO).

Вказівка **FOR** записується у вигляді:

```
FOR ім'я змінної=вираз1 TO вираз2 DO  
Begin  
  вказівка (и);  
End;
```

Читається так: *Для* змінної від виразу1 *До* виразу2 *Виконувати* вказівку(и).

В даному циклі не перевіряється умова закінчення циклу, а просто рахується скільки разів виконується вказівка(и) (зациклювання неможливе) тобто цикл виконується певну кількість разів. Рахунок здійснюється за допомогою спеціального параметра - лічильника. Значення змінної (лічильника) збільшується від виразу1 до виразу2 з кроком 1. Як тільки значення лічильника стає рівним виразу2, виконання тіла циклу припиняється. Крок можна змінити задавши зміну лічильника всередині тіла циклу за допомогою арифметичного виразу.

☞ *Примітка.* Цикл **For** можна зациклити, якщо в тілі циклу перепозначити значення змінної (лічильника) так, щоб воно не стало рівне виразу2.

Розглянемо роботу даної вказівки на прикладі програми.

Вивести на екран всі цілі числа та їх квадрати на проміжку від 10 до 20 включно.

```
Program Kvadrat;  
  var i,s:integer;  
Begin  
  For i:=10 to 20 do  
  begin  
    s:=sqr(i);
```



```
writeln('i=',i:3,' i*i=',s)
end
End.
```

1.4 Вказівка повторення (For - Downto - Do).

Вказівка **For** записується у вигляді:

```
FOR ім'я змінної=вираз1 DOWNTO вираз2 DO
Begin
  вказівка (и);
End;
```

Читається так: *Для* змінної від виразу1 *До* виразу2 *Виконувати* вказівку(и).

Відмінність від попереднього циклу полягає в тому, що значення виразу1 повинно бути більшим значення виразу2. Тобто значення змінної(лічильника) зменшується від виразу1 до виразу2 з кроком -1.

3. Виконання практичного завдання. Інструктаж з техніки безпеки.

а) Розв'язування задач.

В середовищі програмування TurboPascal скласти програми для розв'язку задач:

1. Обчислити $K!$ (190(1))

```
Program z1;
var k, i : integer;
    y : longint;

begin
  readln(k);
  y:=1;
  for i:=1 to k do
    y:=y*i;
  writeln (y);
end.
```

2. Обчислити значення виразу (189(3)).

$$Y = (1+2) + (1+2+3) + (1+2+3+4) + \dots + (1+2+\dots+50)$$

```
Program z2;
var k, i, s, y : integer;

begin
  y:=3;
  k:=0;
  s:=3;
  for i:=3 to 50 do
  begin
    k:=k+i;
    y:=3+k;
    s:=s+y;
```



```
end;  
writeln (s);  
end.
```

3. За даним натуральним значенням змінної n обчислити:

$$Y = \frac{10 \sin(1)}{\sqrt{1}} * \frac{10 \sin(2)}{\sqrt{1 + \sqrt{2}}} * \dots * \frac{10 \sin(n)}{\sqrt{1 + \sqrt{2} + \dots + \sqrt{n}}}$$

```
Program z;  
  Var  y: real;  
      I, n, k : integer;  
Begin  
  Readln(n);  
  Y:=1;  
  I:=1;  
  K:=0;  
  While I<=n do  
    Begin  
      K:= k+sqrt(i);  
      Y:=y*(10*sin(i))/k;  
      I:=I+1  
    End;  
    Writeln('y=',y:6:3)  
  End.
```

б) Самостійна робота (1 учень біля дошки)

1. Обчислити:

$$Y = \sum_{20}^{50} i^2 \quad Y = \prod_{10}^{100} \frac{i+1}{i-2}$$

в) Взаємоперевірка

4. **Домашнє завдання:** Р.5, §2.4, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов.

Додаткова задача:

Дано натуральне число n . Одержати послідовність значень b_i при $i = 1, 2, \dots, n$, якщо відомо, що:

$$a) b_i = \frac{1}{i} \quad б) b_i = \frac{3^i}{(i+1)!}$$

5. **Підсумок уроку.** Узагальнення навчального матеріалу.



Урок № 34

<i>Тема:</i>	Табличні величини (масиви) та їх опис мовою програмування. Алгоритми і програми роботи з табличними величинами.
<i>Мета:</i>	Розглянути табличні величини та навчити описувати їх мовою програмування. Вивчити основні алгоритми роботи з табличними величинами. Розглянути опис головних алгоритмів мовою програмування.
<i>Базові поняття й терміни:</i>	Масив

Хід уроку

1. Актуалізація опорних знань.

- Назвіть базові структури алгоритмів.
- Опишіть їх мовою програмування TurboPascal.
- Зобразіть основні елементи блок-схем.

2. Оголошення теми, мети уроку.

3. Вивчення нового матеріалу

Пояснення вчителя.

При розв'язуванні реальних задач часто виникає необхідність працювати з великою кількістю елементів одного типу, над якими виконуються однакові дії. В таких випадках вигідніше використати табличні величини (масиви) ніж задавати значення цих елементів кожного окремо.

Масив - структурний тип даних, який складається з фіксованого числа елементів одного і того ж типу.

Розрізняють одно- і двовимірні масиви.

Одновимірні масиви

Одновимірний масив - це лінійна сукупність елементів одного типу. Кількість елементів задається під час опису масива і не змінюється на протязі виконання програми.

Розглянемо методи опису масивів.

1. *var ім'я масиву:array[m..n] of тип елементів масиву;*
де m,n - цілі числа.

Наприклад

A:array[1..9] of integer;

C:array[4..12] of real;

2. *type ім'я типу=array[m..n] of тип компонент масиву*
var ім'я масиву:ім'я типу;



Наприклад:

```
type TAB=array[1..9] of integer;  
var A:TAB;
```

При зверненні до елементів масиву необхідно вказувати ім'я масиву і номер елемента (індекс).

Наприклад:

```
A[4], A[8], C[11]
```

Введення і виведення елементів масиву здійснюється поелементно, тому для введення і виведення елементів масиву використовують оператори циклу.

Розглянемо принцип роботи з лінійними масивами на прикладі програми:

Знайти добуток елементів масиву

```
Program DOBUTOK;
```

```
  Var A:array[1..9] of real;  
      d: real;  
      i: integer;
```

```
Begin
```

```
  For i:=1 to 9 do
```

```
  begin
```

```
    writeln('Введіть елементи масиву');
```

```
    write('A[',i,']=');
```

```
    readln(A[i]);
```

```
  end;
```

```
  {Обчислення добутку}
```

```
  d:=1;
```

```
  For i:=1 to 9 do d:=d*a[i];
```

```
  {Вивід добутку на екран}
```

```
  writeln('d=',d)
```

```
End.
```

Двовимірні масиви

Двовимірні масиви є таблицею, яка складається з m -рядків та n -стовпчиків. Двовимірний масив можна замінити сукупністю двох одновимірних масивів.

Описують двовимірний масив майже так само, як і одновимірний.

```
var ім'я масиву:array[m..k,n..h] of тип елементів масиву;
```

Наприклад:

```
var A[1..4,1..9] of real;
```

При зверненні до елементів такого масиву необхідно вказувати два індекси - номер рядка і номер стовпчика.

Опрацювання елементів масиву здійснюється за допомогою двох циклів.



Знайти добуток елементів двовимірного масиву

```

Program DOBUTOK;
  Var A:array[1..9,1..6] of real;
      d: real;
      i,j: integer;
Begin
  For i:=1 to 9 do
  For j:=1 to 6 do
  begin
    write('A[',i,',',j,']=');
    readln(A[i,j]);
  end;
  d:=1;
  For i:=1 to 9 do
  For j:=1 to 6 do
  d:=d*a[i,j];
  writeln('d=',d)
End.

```

3. Формування практичних вмінь і навичок.

Виконання практичного завдання. Розв'язування задач. *Інструктаж з техніки безпеки*).

В середовищі програмування TurboPascal скласти програми для розв'язку задач:

- Скласти програму, яка обраховує середню заробітну плату робітника за рік. Заробітна плата за місяць обчислюється по формулі $z = 300 + \text{random}(50) * \text{random}(m)$, де m змінюється від 1 до 12.

```

Program Z1;
  var  z : array [1..12] of real;
      m: integer;
      sz, s: real;
begin
  for m:=1 to 12 do
    z[m]: =300 + random(50) * random(m);
  s:=0;
  for m:=1 to 12 do
    s:=s+z[m];
  sz:= s/m
  writeln('sz=',sz:0:2);
end.

```

- Скласти програму, яка обраховує силу струму, що проходить через опір $R = 5$ Ом, якщо напругу змінюють від 0 до 20 В з кроком 2 В. Отримані значення сили струму занести в масив та вивести на екран.



```
Program Z2
const r = 5;
var I array [1..20] of real;
    k ,U: integer;
begin
    U:=1;
    K:=1;
    While U<=20 do
        begin
            I[k]:= U/R;
            Writeln('U=',U:0,'B',' I=',I:0:2,'A');
            k:=k+1;
            u:=u+2;
        end;
    end.
```

4. **Домашнє завдання:** Р.5, §2.8, 2.9, Ч.2 "Інформатика". І.Т. Зарецька, А.М. Гуржій, О.Ю.Соколов.

Підготуватись до тематичного оцінювання.

5. **Підсумок уроку.** Оцінювання учнів.



Урок № 35

<i>Тема:</i>	Тематичне оцінювання з теми: "Основи алгоритмізації та програмування"
<i>Мета:</i>	Перевірити засвоєння учнями теоретичного матеріалу з теми: "Основи алгоритмізації та програмування", вміння створювати програми з використанням вказівок мови програмування та використовувати середовище програмування для створення та виконання програм.

Хід уроку

1. Організаційний момент.

2. Проведення тематичного оцінювання. *Інструктаж з техніки безпеки.*

Завдання для тематичного оцінювання містять, як теоретичні так практичні питання. Розв'язки задач оформити в середовищі Turbo Pascal, назви програм згідно номера комп'ютера та рівня, наприклад: Z01_R4, відповіді на теоретичні питання записати в зошит.

Завдання для тематичного оцінювання

Початковий рівень

1. Записати означення алгоритму та перерахувати його властивості.
2. Запишіть відомі вам типи величин.
3. Запишіть відомі вам вказівки розгалуження та повторення.
4. Скласти програму, яка додає два числа та виводить отриманий результат на екран.

Середній рівень

1. Запишіть, хто може бути виконавцем алгоритму, програми. Відповідь поясніть.
2. Поясніть різницю між вказівками: `write(x:0:2)`, `writeln(x:6:3)`;
3. Запишіть, як описуються змінні X - цілого типу, та Y - дійсного типу в розділі опису змінних.
4. Скласти програму, яка обраховує площу та периметр квадрата з стороною – A.

Достатній рівень

1. Що розуміють під терміном масив, які типи масивів ви знаєте?
2. Які вказівки відносять до вказівок розгалуження? Відповідь поясніть.
3. Скласти програму для обчислення площі довільного трикутника, якщо відомо його сторони. Забезпечити перевірку правильності початкових даних.



4. Скласти програму для обчислення значення функції $Y=3x^2-\sin(2x)$, якщо x змінюється від 0 до 10 з кроком 0,1. Результат записати в масив та вивести на екран.

Високий рівень

1. Які вказівки повторення ви знаєте, в чому полягає різниця між ними?
2. Опишіть метод пошуку мінімального елемента в одновимірному масиві.
3. Скласти програму для обчислення $S=1/1!+1/2!+\dots+1/k!$
4. Скласти програму для обчислення значення виразу:

$$Y = \begin{cases} \frac{x+2}{x-4} + 10x, & -\infty < x \leq 10 \\ x^2 - \sqrt{x-20}, & 10 < x < 20 \\ \frac{\sqrt{25x-20} + 2}{x+5}, & 20 \leq x < +\infty \end{cases}$$

3. Підсумок уроку.

4. **Домашнє завдання:** Повторити базові структури алгоритму.



Додаток.

1. Скласти програму, яка обраховує значення функції $Y=2X^2$, якщо x змінюється від -5 до 5 з кроком 1.

{Вказівка повторення з передумовою - WHILE}

```

Program z1;
  var x:real;           {змінна X дійсного типу - аргумент}
      y:real;           {змінна Y - дійсного типу - функція, результат}
begin                 {початок програми}
  x:=-5;              {встановлюємо початкове значення змінної - X}
  writeln(' X | Y');  {виводимо на екран підказку}
  while x<=5 do      {ПОКИ X<=5 виконувати}
    begin           {початок циклу}
      y:=2*sqr(x);  {обраховуємо значення функції Y за заданою формулою}
      writeln(x:0:2,' ',y:0:5); {виводимо значення X та Y на екран}
      x:=x+1;       {збільшуємо значення змінної X на 1}
    end;           {кінець вказівки повторення (циклу)}
end.               {кінець програми}

```

2. Скласти програму, яка обраховує значення функції $Y=2X^2$, якщо x змінюється від -5 до 5 з кроком 1.

{Вказівка повторення з післяумовою - REPEAT}

```

Program z1;
  var x:real;
      y:real;
begin
  x:=-5;              {- початкове значення}
  writeln(' X | Y');
  repeat             {- повторювати групу команд}
    y:=2*sqr(x);
    writeln(x:0:1,' ',y:0:5);
    x:=x+1;          {- крок зміни X}
  until x>5;         {- до тих пір поки X не стане більшим за 5 (кінцеве значення)}
end.

```

3. Скласти програму, яка обраховує значення функції $Y=2X^2$, якщо x змінюється від -5 до 5 з кроком 1.

{Вказівка повторення з параметром - FOR}

```

Program z1;
  var x:integer; {при використанні вказівки повторення з параметром лічильник завжди цілого типу}
      y:real;
begin
  writeln(' X | Y');
  for x:=-5 to 5 do {- Для X від -5 до 5 з кроком 1 виконувати групу команд}
    begin         {при використанні вказівки повторення з параметром rhjr pfd;lb - 1}
      y:=2*sqr(x);
      writeln(x:2,' ',y:0:5);
    end;
end.

```




7. Вивести таблицю значень функції $y = x^4 - 5x^3 - 2x^2 + 0.1$ для значень x , які змінюються в проміжку від -5 до 3 з кроком 0.2 .

```

Program z4_37;                                {Назва програми}
  var x,y:real;                                {Опис змінних X та Y}
begin                                          {Початок програми}
x:=-5;                                        {Початкове значення змінної X}
repeat                                       {Повторювати - початок циклу (вказівки повторення)}
  y:=X*X*X*X-5*X*X*X-2*SQR(X)+0.1;         {Обрахунок функції Y}
  Writeln('X=',x:0:1,' Y=',Y:0:3);       {Вивід на екран значень змінної X та функції Y}
  x:=x+0.2;                                 {Зміна значення змінної X з заданим кроком =
                                           0.2}
until x>3.1;                                {поки X не стане більше 3 - кінець циклу
                                           (вказівки повторення)}
end.                                          {кінець програми}

```

8. Скласти програму, яка виводить на екран суму значень функції $Y=1+1/(n*n)$, якщо аргумент функції - N змінюється від 1 до 100 з кроком 1 .

```

Program z1;
  var n:integer;
      s,y:real;
begin
  n:=1;
  s:=0;                                       {початкове значення суми}
  while n<=100 do
  begin
    y:=1+1/sqr(n);                           {Обрахунок значень заданої функції}
    s:=s+y;                                   {Обрахунок суми значень заданої функції}
    n:=n+1;                                   {зміна значень аргументу функції з заданим кроком}
  end;
  writeln('s=',s:0:2);
end.

```

```

Program z1;
  var n:integer;
      s,y:real;
begin
  s:=0;
  for n:=1 to 100 do
  begin
    y:=1+1/sqr(n);
    s:=s+y;
  end;
  writeln('s=',s:0:2);
end.

```

9. Скласти програму, яка підраховує суму чисел, які знаходяться в файлі $Z.in$. Вивести отримане значення в файл $Z.out$.

```

Program z1;
  var f,f1: text;                             {файлові змінні f,f1 типу text}
      ch,s: real;
begin
  assign(f,'z.in');                          {пов'язує файловою змінною f з файлом z.in}
  reset(f);                                  {відкриває файл f для читання}
  s:=0;                                       {початкове значення суми}
  while not(eof(f)) do                       {поки не має кінця файлу f виконувати групу команд}
  begin
    read(f,ch);                              {послідовно читає з файлу f значення чисел в змінну ch}
    s:=s+ch;                                 {підраховує суму зчитаних чисел}
  end;
end.

```



```

end;
close(f);           {закриває файл f}
assign(f1,'z.out'); {пов'язує файловою змінною f1 з файлом z.out}
rewrite(f1);        {відкриває файл f1 для запису інформації}
writeln(f1,'s=',s:0:2); {вивід результату в файл f1}
close (f1);         {закриття файлу f1}
end.

```

10. Скласти програму, яка підраховує суму чисел, які знаходяться в файлі Z.in. Вивести отримане значення на екран, та в файл Z.OUT.

```

Program z1;
var f,f1: text;      {файлові змінні f,f1 типу text}
    ch,s: real;      {ch- числа, які зчитуються з файлу, S- сума зчитаних чисел}
begin
assign(f,'z.in');    {пов'язує файловою змінною f з файлом z.in}
reset(f);            {відкриває файл f для читання}
s:=0;                {початкове значення суми}
while not(eof(f)) do {поки не має кінця файлу f виконувати групу команд}
begin
    read(f,ch);      {початок вказівки повторення}
    s:=s+ch;         {послідовно читає з файлу f значення чисел в змінну ch}
end;                 {підраховує суму зчитаних чисел}
close(f);            {кінець вказівки повторення}
assign(f1,'z.out');  {закриває файл f}
rewrite(f1);         {пов'язує файловою змінною f1 з файлом z.out}
writeln(f1,'s=',s:0:2); {відкриває файл f1 (z.out) для запису інформації}
close (f1);         {вивід результату в файл f1 (z.out)}
writeln('s=',s:0:2); {закриття файлу f1 (z.out)}
                    {Виводить отримане значення суми на екран}
end.

Program z;
var f:text;          {F - файлова змінна типу текст}
    a,b,s,p: real;   {a,b,s,p - змінні дійсного типу}
begin
assign(f,'r.dat');   {початок програми}
reset(f);            {пов'язує файловою змінною f з файлом r.dat}
while not(eof(f)) do {відкриває файл f (r.dat) для читання}
{Поки не має кінця файлу виконувати - початок вказівки по-
вторення}
begin
    read(f,a,b);     {початок циклу}
    s:=a*b;          {послідовно зчитує з файлу f (r.dat) по два числа - a і b}
    p:=2*(a+b);      {обрахунок площі прямокутника}
    writeln('s=',s:0:2,' p=',p:0:2); {обрахунок периметра прямокутника}
                    {виводить значення отриманих - площі і
                    периметра на екран}
end;                 {кінець вказівки повторення (циклу)}
close(f);            {закриває файл f (r.dat)}
end.                 {кінець програми}

```

11. Скласти програму, яка обчислює суму, добуток і кількість додатних елементів масиву A(5).

```

Program z1;
var a: array [1..5] of real; {Назва програми}
    s,d: real;                {масив A на 5 елементів дійсного типу}
    k,nom: integer;          {сума і добуток - дійсного типу}
begin
    {кількість і номер елемента - цілого типу}
    {початок програми}
    {блок введення елементів масиву в пам'ять комп'ютера}
    for nom:=1 to 5 do
    begin
        write('A[' ,nom, ']='); {змінюємо номер елемента від 1 до 5 з кроком 1}
        readln(a[nom]);        {початок циклу}
                                {вивід підказки- A[1]=}
                                {введення значення елемента масиву в
                                пам'ять}
    end;
                                {кінець циклу}
end.

```



```

{Обрахунок}
s:=0;
k:=0;
d:=1;
for nom:=1 to 5 do
1}
begin
  if a[nom]>0 then
    begin
      s:=s+a[nom];
      k:=k+1;
      d:=d*a[nom];
    end;
end;
{виведення результатів на екран}
if k:=0 then d:=0;

writeln('S=',s:0:2);
writeln('K=',k:0);
writeln('d=',d:0:2);

end.

```

{Початкове значення суми}
{Початкове значення кількості}
{Початкове значення добутку}
{змінюємо номер елемента від 1 до 5 з кроком 1}
{початок циклу}
{Якщо елемент більший за нуль (додатний) то}
{початок вказівки розгалуження}
{суму збільшуємо на значення елемента}
{до кількості додаємо 1}
{добуток множимо на значення елемента}
{кінець вказівки розгалуження}
{кінець циклу}
{якщо кількість додатних елементів в масиві рівна 0 то добутку присвоїти 0}
{вивід отриманого значення суми}
{вивід отриманого значення кількості}
{вивід отриманого значення добутку}
{кінець програми}

12.Скласти програму, яка обчислює суму, добуток і кількість додатних елементів масиву A(3,5).

```

Program z1;
var a: array [1..3,1..5] of real;
    s,d: real;
    k,x,y: integer;
цілого типу}
begin
  {блок введення елементів масиву в пам'ять комп'ютера}
  for x:=1 to 3 do
  for y:=1 to 5 do
ком 1}
  begin
    write('A[' ,x, ',' ,y, '=');
    readln(a[x,y]);
  пам'ять}
  end;
  {Обрахунок}
  s:=0;
  k:=0;
  d:=1;
  for x:=1 to 3 do
  for y:=1 to 5 do
  begin
    if a[x,y]>0 then
      begin
        s:=s+a[x,y];
        k:=k+1;
        d:=d*a[x,y];
      end;
    end;
  {виведення результатів на екран}
  if k:=0 then d:=0;

  writeln('S=',s:0:2);
  writeln('K=',k:0);
  writeln('d=',d:0:2);
  end.

```

{Назва програми}
{масив A на 15 елементів дійсного типу}
{сума і добуток - дійсного типу}
{кількість, номери рядків і стовпчиків - цілого типу}
{початок програми}
{змінюємо номер рядка від 1 до 3 з кроком 1}
{змінюємо номер стовпчика від 1 до 5 з кроком 1}
{початок циклу}
{вивід підказки- A[1]=}
{введення значення елемента масиву в пам'ять}
{кінець циклу}
{Початкове значення суми}
{Початкове значення кількості}
{Початкове значення добутку}
{змінюємо номер рядка від 1 до 3 з кроком 1}
{змінюємо номер стовпчика від 1 до 5 з кроком 1}
{початок циклу}
{Якщо елемент більший за нуль (додатний) то}
{початок вказівки розгалуження}
{суму збільшуємо на значення елемента}
{до кількості додаємо 1}
{добуток множимо на значення елемента}
{кінець вказівки розгалуження}
{кінець циклу}
{якщо кількість додатних елементів в масиві рівна 0 то добутку присвоїти 0}
{вивід отриманого значення суми}
{вивід отриманого значення кількості}
{вивід отриманого значення добутку}
{кінець програми}



Зміст

ПЕРЕДМОВА.....	3
Календарно-тематичне планування	4
Урок № 1.....	6
Поняття про комп'ютерні презентації. Знайомство з програмою Microsoft Power-Point.	6
Урок № 2.....	11
Розмітка та оформлення слайдів. Тексти та малюнки в слайдах.	11
Урок № 3.....	15
Імпортування об'єктів, створених в середовищі інших програмних засобів. <i>Вставка</i> діаграм, таблиць, колонтитулів, відеокліпів в слайд. Анімація та звук в слайдах.	15
Урок № 4.....	18
Інтерактивні презентації. Демонстрація презентацій.....	18
Урок № 5.....	20
Узагальнення і систематизація знань.	20
Урок № 6.....	21
Тематичне оцінювання з теми "Комп'ютерні презентації".	21
Урок № 7.....	22
Поняття про електронні таблиці. Табличний процесор. Основні об'єкти ЕТ.....	22
Урок № 8.....	26
Введення та редагування даних. Робота з книгами.	26
Урок № 9.....	29
Форматування електронних таблиць.....	29
Урок № 10.....	32
Виконання обчислень в електронних таблицях. Робота з формулами. ..	32
Урок № 11.....	36
Використання функцій в ЕТ.....	36
Урок № 12.....	39
Діаграми в електронних таблицях.	39
Урок № 13.....	44
Упорядкування, фільтрація та пошук даних.....	44
Урок № 14.....	47
Виведення табличних даних та діаграм на друк. Узагальнення і систематизація знань.....	47
Урок № 15.....	49
Тематичне оцінювання з теми "Табличний процесор"	49
Урок № 16.....	50
Бази даних. Системи управління базами даних. Основні об'єкти баз даних.	50



Урок № 17.....	53
Створення баз даних. Таблиці.....	53
Урок № 18.....	56
Редагування даних. Форми.....	56
Урок № 19.....	59
Поля підстановки. Створення зв'язків між таблицями. Ключові поля... ..	59
Урок № 20.....	62
Сортування, фільтрація і пошук даних.	62
Урок № 21.....	64
Запити.....	64
Урок № 22.....	67
Звіти.	67
Урок № 23.....	70
Тематичне оцінювання з теми "Бази даних. Системи управління базами даних".....	70
Урок № 24.....	76
Поняття інформаційної моделі. Поняття технології комп'ютерного моделювання. Інформаційні моделі. Побудова моделі. Основні етапи комп'ютерного моделювання.	76
Урок № 25.....	80
Алгоритми. Властивості алгоритмів. Форми подання алгоритму. Виконавець алгоритму.	80
Урок № 26.....	82
Базові структури алгоритмів: слідування, розгалуження, повторення. Графічні схеми базових структур алгоритмів.	82
Урок № 27.....	87
Порядок складання алгоритмів. Поняття програми. Конструювання алгоритмів за методом "зверху донизу".....	87
Урок № 28.....	89
Навчальна алгоритмічна мова. Правила опису алгоритмів навчальною алгоритмічною мовою.	89
Урок № 29.....	92
Поняття програми. Мова програмування. Переклад програм з однієї мови на іншу. Поняття про транслятори, систему програмування. Поняття про середовище програмування TurboPascal.	92
Урок № 30.....	97
Величини. Основні характеристики величин. Вказівка присвоювання.....	97
Урок № 31.....	101
Поняття основного та допоміжного алгоритму. Процедури та функції. Формальні та фактичні параметри. Глобальні та локальні змінні. Розв'язування задач.	101



Урок № 32.....	105
Вказівка розгалуження та вибору, їх опис навчальною алгоритмічною мовою та мовою програмування. Розв'язування задач з використанням вказівки розгалуження.....	105
Урок № 33.....	109
Вказівка повторення та її опис навчальною алгоритмічною мовою і мовою програмування. Розв'язування задач з використанням вказівки повторення.	109
Урок № 34.....	114
Табличні величини (масиви) та їх опис мовою програмування. Алгоритми і програми роботи з табличними величинами.....	114
Урок № 35.....	118
Тематичне оцінювання з теми: "Основи алгоритмізації та програмування"	118
Зміст	125
Список літератури.....	128

**Список літератури.**

1. Верлань А.Ф., Апатова Н.В. Інформатика (підручник). —К.: Форум, 2001.
2. В.Д. Руденко, О.М. Макарчук, М.О. Патланжоглу Курс інформатики. —К.:Фенікс, 2001.
3. Валецька Т.М. Комп'ютерні мережі. —К., 2004.
4. Зарецька І.Т., Гуржій А.М., Соколов О.Ю.. Інформатика Ч.1,2 —К.: Форум, 2004.
5. Зарецька І.Т., Гуржій А.М., Соколов О.Ю.. Методичний посібник з інформатики —Харків: Факт, 2004.
6. Гаєвський О.Ю. Інформатика 7-11 клас — К.: А.С.К., 2005.
7. Матвієнко Ю.С. Лабораторний практикум з інформатики для 10 класу —Полтава, 2005.
8. Білик О.О., Заячковський В.М., Пойда С.А. Збірник практичних робіт з інформатики (10-11 клас) – Вінниця: Вінницький обласний інститут післядипломної освіти педагогічних працівників, 2003. – 80 с.
9. Могильна Н.М., Створення презентацій засобами Microsoft Power-Point — Ріпкинська гімназія, 2005.
10. Оліференко Н.В. Основи інформатики у визначеннях, таблицях і схемах — Х.: Ранок, 2006.
11. Жалдак М.І., Морзе Н.В., Мостіпан О.І. Інформатика. програми для загальноосвітніх навчальних закладів. – Запоріжжя: Прем'єр, 2003.
12. Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачі по програмуванню. – М.: Наука, 1988. – 224 с.
13. Абрамов С.А., Зима Е.В. Начала програмування на языке паскаль. – М.: Наука, 1987.
14. Комплект плакатів з інформатики для 10-11 класів. – Х.: Веста: Вид-во "Ранок", 2006.



ПРАВИЛА РОБОТИ УЧНІВ У КОМП'ЮТЕРНОМУ КЛАСІ

Файл Редагування Перегляд Вибране Інструменти Допомога

Безперервна робота учнів з екраном відеомонітора не повинна перевищувати: *

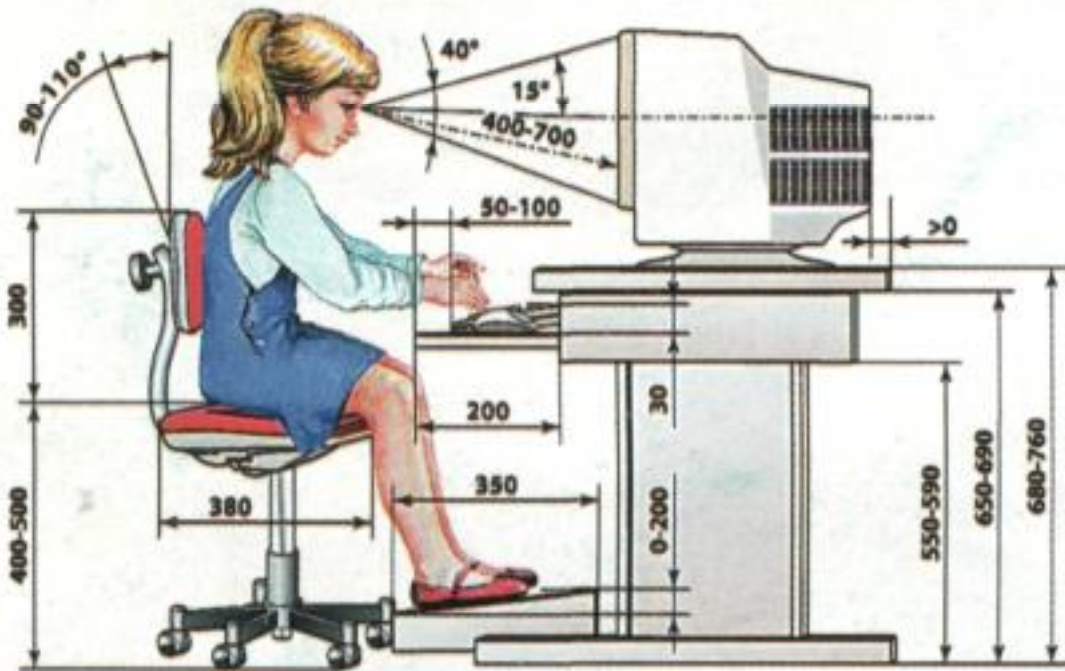
для учнів I класу (6 років) — 10 хв за одну навчальну годину;
 для учнів II–V класів — 15 хв за одну навчальну годину;
 для учнів VI–VII класів — 20 хв за одну навчальну годину;
 для учнів VIII–IX класів — 25 хв за одну навчальну годину;
 для учнів X–XII класів та студентів вищих навчальних закладів I та II рівнів акредитації — на першій годині занять — 30 хв, на другій годині — 20 хв.

* Вимоги згідно з Постановою Кабінету Міністрів України від 11.03.2004 № 1042/2004-В «Про затвердження правил організації навчально-виховного процесу в загальноосвітній школі».

Вимоги безпеки в аварійних ситуаціях

- ⚠ При появі незвичайного звуку або вимкненні апаратури потрібно негайно припинити роботу й довести це до відома вчителя або лаборанта.
- ⚠ При появі запаху паленого сім'я припинити роботу, вимкнути апаратуру і повідомити про це вчителя або лаборанта.
- ⚠ При потрапленні людини під напругу необхідно знайти відповідне робоче місце, надати першу допомогу та викликати швидку допомогу.
- ⚠ При вимкненні потужні необхідно знайти клас, викликати пожежну команду і приступити до гасіння пожежі засобами, які є.
- ⚠ У разі недотримання учнями вимог з безпеки при роботі з комп'ютером адміністрація школи може притягнути порушників до дисциплінарної та адміністративної відповідальності.

Організація комп'ютеризованого робочого місця



Комплекс вправ для очей

Варіант I



1. Погляд спрямувати вліво-вправо, вгору-вниз, вперед-назад без затримки в певному положенні. Повторити 5 разів в одному і 5 разів у зворотньому напрямку.
2. Закрити очі на різному відстані, моргнути очі і подивитися на кінчик носа на різному відстані.
3. Колові рухи очей: до 4 разів вліво і вправо.

Вихідне положення — сидіти (відвернувшись від комп'ютера), спина прямо, очі відкриті, погляд — прямо.



1. Швидко мигнути очима 15 с.
2. Заплющити очі. Не відкриваючи очей, начебто піднявшись ліворуч на рукою край-сторінку, повернутися у вільне положення. Та само піднявшись праворуч на рукою край-сторінку і повернутися у вільне положення. Повторити 5 разів.
3. Спокійно поглядіть із закритими очима, розслабляючись протягом 5 с.

Комплекс вправ для зняття м'язового напруження



Вихідне положення — стоячи, ноги разом, руки вниз.



1. Прямі руки розвести в бок до локтя долонь долонь, зробити вдих.



2. Схрестити руки перед грудьми, міцно обхватити себе за плечі, повторити 3 рази.



3. Колові рухи літкими вперед протягом 5 с.



4. Ті саме назад. Дихати рівномірно.



ЕЛЕКТРОННІ ТАБЛИЦІ

Файл Редагування Перегляд Вибране Інструменти Допомога

Електронні таблиці (ЕТ) –

структура даних у пам'яті комп'ютера у вигляді двовимірної таблиці для введення, зберігання, обробки даних у її комірках та виведення результатів розрахунків у зручному вигляді

Табличний процесор –

прикладна програма для опрацювання ЕТ

Основна властивість ЕТ

Зміна початкових даних → Пере-рахування за формулами → Новий стан ЕТ

Об'єкти табличного процесора

- файл (книга)
- аркуш
- електронна таблиця
- діаграма

Приклад

Дані для побудови діаграми (електронна таблиця)

	A	B	C	D	E	F	G	H
1	Дослідження учнів							
2	Рівень	Кількість учнів						
3	високий	7						
4	достатній	12						
5	середній	14						
6	низький	2						

Об'єкти електронної таблиці

Назва об'єкта	Комірка (клітинка)	Стовпець	Рядок	Діапазони
Позначення (за замовченням)	A7, B2, DC12345 (складається з імені стовпчика та номера рядка)	A, B, C, ..., AA, AB, ... (літери латинського алфавіту та їх комбінації)	1, 2, 11, ..., 5267... (арабські цифри)	
Відображення (приклад)				

Дані електронних таблиць

	A	B	C
1	№ з/п	Прибуток	Дата
2	1	1020,50	01.05.06
3	2	345,25	27.08.06
4	3	=B2+2+МАКС(B1:B2)	

Текст
Числові дані
Формула

Типи числових даних ЕТ

- Цілі (5; 12 ...)
- Дійсні
 - грошовий, фінансовий (10,0 грн ...)
 - відсотковий (100%, ...)
 - дробовий (2 1/2)
 - експоненціальний (1,03E + 0,1 ...)
- Дата та час (19 січ 2007, 09:00 ...)

Склад формули в ЕТ

Знак рівності	Адреси комірок, знаки арифметичних операцій, функції (математичні, логічні, статистичні тощо) діапазони комірок, числа
=	B2 + 2 + МАКС (B1 : B2)



БАЗИ ДАНИХ

Файл Редагування Перегляд Вибране Інструменти Допомога

Інформаційно-пошукові системи —

сукупність баз даних та комплексу засобів для створення, зберігання, зміни та пошуку необхідної інформації

Бази даних (БД) —

сукупність даних предметної галузі, яким властива структурованість і взаємозв'язаність, а також незалежність від прикладних програм

+
 =

Інформаційно-пошукова система

База даних Засоби створення, зберігання та пошуку інформації

Класифікації БД

За характером інформації, що зберігається

- Фактографічні [картотеки]
- Документальні [архіви]

За способом зберігання даних

- Централізовані
- Розподілені

За структурою організації даних (моделі БД)

Ієрархічна

```

      graph TD
        A1((A1)) --- B1((B1))
        A1 --- B2((B2))
        B1 --- C1((C1))
        B1 --- C2((C2))
        B1 --- C3((C3))
        B2 --- C4((C4))
        B2 --- C5((C5))
        B2 --- C6((C6))
      
```

Мережна

```

      graph TD
        A1((A1)) --- B1((B1))
        A1 --- B2((B2))
        A2((A2)) --- B3((B3))
        A2 --- B4((B4))
        B1 --- C1((C1))
        B1 --- C2((C2))
        B2 --- C3((C3))
        B3 --- C4((C4))
        B4 --- C5((C5))
      
```

Реляційна (від латин. relatio — відносини)

Об'єкти реляційної БД

Поле БД —

найпростіший об'єкт БД (стовпчик таблиці) для зберігання значення одного параметра (властивості) реального об'єкта

Запис БД —

сукупність логічно зв'язаних полів (рядок таблиці), що містить дані про один реальний об'єкт

Ключове поле БД (ключ БД) —

одне або декілька полів, що однозначно визначають записи

Ключове поле БД

↓

Поля

Т	Код видання	Автор	Назва	Видавництво
а	1091	Роганін О.М.	Геометрія 10 кл.	Ранок
б	2135	Клімов А.А.	Міфи і легенди	Ранок
л	0618	Конан Дойль	Загублений світ	Ранок
и	2136	Клімов А.А.	Релігія світу	Ранок
ц	2346			
я				

Записи БД

Етапи створення БД

- 1 Проектування БД
- 2 Створення комп'ютерної структури БД
- 3 Введення даних БД

Засоби створення комп'ютерної БД

- Алгоритмічні мови (Pascal, QBasic)
- За допомогою Систем Управління Базами Даних (СУБД) (Access)
- Візуальні середовища (VBASIC, Delphi)

СУБД —

прикладне програмне забезпечення для створення БД, зберігання, упорядкування та пошуку даних

Основні об'єкти СУБД Access

- таблиці
- запити
- форми
- звіти
- макроси
- модулі



ОСНОВИ АЛГОРИТМІЗАЦІЇ

Файл Редагування Перегляд Вибране Інструменти Допомога

Алгоритм –

це скінченна однозначна послідовність точно визначених дій для виконавця, яка забезпечує досягнення визначеної мети або розв'язання поставленої задачі.

Виконавець –

людина, група людей, тварина або технічний пристрій, що може виконувати команди алгоритму

Схема функціонування виконавця алгоритму

```

    graph LR
      A[Система команд виконавця] --> B[Дані]
      A --> C[Алгоритм]
      B --> D[Виконавець]
      C --> D
      D --> E[Результат]
      style D fill:#fff,stroke:#000,stroke-width:2px
      style D image:executor
      
```

Властивості машинних алгоритмів

- дискретність
- детермінованість
- формальність
- скінченність
- результативність
- масовість

Форми подання алгоритму

Словесна

Дано два числа. Якщо перше число більше від другого, то обчислити їх суму, інакше – обчислити різницю. Вивести результат обчислення.

Словесно-формульована

- Ввести змінні a, b .
- Перевірити: якщо $a > b$, то знайти $c = a + b$, інакше $c = a - b$.
- Вивести значення змінної c .

У вигляді блок-схеми

```

    graph TD
      Start([початок]) --> Input[/введення a, b/]
      Input --> Decision{a > b}
      Decision -- так --> Calc1[c = a + b]
      Decision -- ні --> Calc2[c = a - b]
      Calc1 --> Output[/виведення c/]
      Calc2 --> Output
      Output --> End([кінець])
      
```

Мова програмування

```

program Z1;
var a, b, c : Integer;
begin
  write ('введіть a, b');
  read (a, b);
  if a > b then c = a + b
  else c = a - b;
  write ('c = ', c);
end.
    
```

Базові структури алгоритмів

Слідування

Лінійний алгоритм – алгоритм, в якому використовується тільки структура «слідування»

Розгалуження (вибір)

повна форма

неповна форма

Алгоритм з розгалуженням – алгоритм, в основі якого лежить структура «розгалуження»

Повторення (цикл)

цикл з передумовою

цикл з післяумовою

Циклічний алгоритм – алгоритм, в основі якого лежить структура «повторення»



ОСНОВНІ ПОНЯТТЯ МОВИ ПРОГРАМУВАННЯ

Файл Редагування Перетяг Видране Інструменти Допомога

Програма —
запис алгоритму мовою програмування

Мова програмування —
система позначень для точного опису алгоритмів для виконання комп'ютером

```

program Summa;
var a, b, sum (integer);
begin
writeLn ('введіть 2 числа');
read (a, b);
sum := a + b;
writeLn ('Сума чисел = ', sum);
end.

```

Класифікація мов програмування

Мови програмування високого рівня

- алгоритмічні (процедурні)
 - процедурні (Fortran, Basic, Pascal)
 - об'єктно-орієнтовані (Delphi)
- дескриптивні (непроцедурні)
 - функціональні (Lisp)
 - логічні (Prolog)

Мови програмування низького рівня

Машинні коди

Види трансляторів

Інтерпретатор
транслокує в машинні коди кожну **вказівку** програми окремо, яку процесор і виконує

Компілятор
транслокує в машинні коди **всю програму**, яку потім процесор виконує

програма

Вказівка 1

Вказівка 2

⋮

Вказівка n

Вказівка n

↓

Вказівка 2

↓

Вказівка 1

1001010110 **Машинні коди** 101010010

Характеристики мови програмування

Алфавіт

сукупність символів, які дозволено використовувати при створенні програми

- літери латинського алфавіту (A – Z, a – z)
- десятичні цифри (0, ... 9)
- спеціальні символи (+, -, *, /, @, \$, #, ...)
- комбінаційні символи (<=, >, < > ...)
- службові слова (if, for, end ...)

Синтаксис

сукупність правил та вимог побудови вказівок алгоритмів мовою програмування

Семантика

пояснення смислових значень кожного елемента та дій комп'ютера при їх виконанні

Основні поняття мови програмування

- ідентифікатори (імена об'єктів)
- елементи даних (константи, змінні)
- типи даних
 - прості (цілі, дійсні, символьні, логічні)
 - складні (масиви, режими, записи, вказівки, файли, об'єкти)
- оператори (команди)
 - прості (присвоєння, введення, виведення)
 - складні (цикли, розгалуження)
- операції в даних
 - арифметичні (+, -, *, /)
 - операції відносин (>, <=, >, < > ...)
 - логічні (&, або, &)

Приклад опису розв'язання задачі

Алгоритмічна мова	Basic	Pascal
int sum	REAL sum	program Summa;
int a, b		var a, b, sum (integer);
int sum	INPUT "Введіть 2 числа" a, b	begin
sum := a + b	sum := a + b	writeLn ('введіть 2 числа');
виведення sum	PRINT "Сума чисел = " sum	read (a, b);
endif	END	sum := a + b;
		writeLn ('Сума чисел = ', sum);
		end.