**УРОК №24.**

**ТЕМА:** ***Поняття підпрограми та її аргументів. Поняття локальних і глобальних змінних.***

***Формальні і фактичні параметри підпрограм.***

**Тема:** **Поняття підпрограми та її аргументів. Поняття локальних і глобальних змінних. Формальні і фактичні параметри підпрограм.**

**Мета:** Актуалізувати знання учнів про прапорці та групи перемикачів, реалізацію розгалужень з використанням прапорців. Подати новий навчальний матеріал про підпрограми та їх аргументи, локальні та глобальні змінні, формальні та фактичні параметри підпрограм. Закріпити даний матеріал під час виконання практичних завдань.

**Тип уроку:** Комбінований урок.

**План уроку:**

Актуалізація опорних знань учнів.

Вивчення нового навчального матеріалу.

1. Підпрограми. Види підпрограм.
2. Структурний опис процедури. Вхідні та вихідні параметри.
3. Структура, опис підпрограми-функції. Вхідні і вихідні параметри.

Закріплення знань.

Домашнє завдання.

**Актуалізація опорних знань:**

**1.** Для чого використовують прапорці та перемикачі?

**2.** Що визначає властивість **Checked** для перемикачів і прапорців?

**3.** Поясніть відмінності використання команд розгалуження при складанні проектів з прапорцями та перемикачами.

**Вивчення нового навчального матеріалу:**

**1. Підпрограми. Види підпрограм.**

Підпрограми призначенні для реалізації алгоритмів опрацювання окремих частин деякої складної задачі.

В одній і тій ж програмі окремі її частини можуть виконуватися одноразово, а інші – багаторазово.

***Підпрограми доцільно використовувати тоді, коли в різних місцях програми необхідно виконувати одну і ту ж саму послідовність дій.*** Використання підпрограм полегшує програмування задач, скорочуючи час на її розробку і зменшуючи зусилля. Отже, ***підпрограма – це програма всередині програми.***

Кожна програма може складатись із великої кількості різних підпрограм.

Залежно від прийнятих правил оформлення і використання розрізняють різні ***типи підпрограм:***

***– підпрограми-процедури;***

***– підпрограми-функції.***

Принципова ***відмінність між ними полягає в тому, що функція повертає значення у місце виклику, а процедура – ні.***

**2.** **Структурний опис процедури. Вхідні та вихідні параметри.**

***Загальний опис процедур:***

***procedure <ім’я> (<список формальних параметрів>);***

***<розділи описів і оголошень процедури>***

***begin***

***<розділ команд процедури>***

***end;***

Як видно із заголовка, процедура може бути і без параметрів. У цьому випадку вказується тільки ім’я процедури.

***Список формальних параметрів – це перелік імен параметрів (змінних) разом із зазначенням їхніх типів.***

Правила запису формальних параметрів:

Параметри відокремлюються один від одного комою. Перед типом ставиться двокрапка. Якщо декілька параметрів мають один тип, то він зазначається для усієї групи параметрів. У цьому випадку групи параметрів різних типів відділяються одна від одної крапкою з комою.

За допомогою формальних параметрів здійснюються передавання фактичних даних у процедуру а також результатів виконання процедур у програму, що її викликала.

***Види формальних параметрів:***

***- параметри-аргументи (або параметри-значення) – це вхідні данні для процедури;***

***- параметри-результати (параметри-зміні) – параметри, через які можна повертати результати роботи процедури в основну програму.***

Як їх відрізнити?

Перед списком параметрів-результатів кожного типу записують слово *var*. Параметри-аргументи оголошуються без нього.

Розглянемо приклад процедури з назвою *Cina*, яла визначає ціну *k* грамів цукерок, якщо вартість 1кг дорівнює 28грн. 60коп.

*procedure Cina (k:integer; var c:real);*

*begin*

*c:=k\*28.6;*

*end;*

у наведеному прикладі *k* – формальний параметр-аргумент, c – формальний параметр-результат.

До процедури звертаються з розділу команд основної програми або іншої підпрограми. ***Процедуру викликають за допомогою команди виклику:***

***<ім’я процедури> (<список фактичних параметрів>);***

***Фактичними називають параметри, які записують у команді виклику процедури. Фактичні параметри теж бувають параметрами-аргументами і параметрами-результатами.*** Фактичними параметрами-аргументами можуть бути сталі, зміні, вирази, а параметрами-результатами лише зміні. Типи даних в команді виклику процедури не зазначають.

Між фактичними і формальними параметрами має бути *відповідність* за кількістю і типами, тобто:

- кількість формальних і фактичних параметрів повинна бути однакова;

- тип кожного фактичного параметра повинен збігатися з типом відповідного формального параметра.

Проте відповідні фактичні та формальні параметри можуть мати різні імена.

*Виклик процедури здійснюється так:* значення фактичних параметрів присвоюється відповідним формальним параметрам процедури, виконується процедура, визначаються параметри-результати, значення яких надаються (повертаються) відповідними фактичними параметрами у команді виклику.

Зупинимося на одному правилі описання і використання змінних.

***Змінні, описані в основній програмі, називаються глобальними. Вони можуть використовуватися, як в основній програмі, так в усіх підпрограмах. Змінні, описані усередині процедур і функцій, називаються локальними.*** Вони можуть використовуватися тільки усередині даної процедури (або функції) і не можуть використовуватися в основній програмі.

Таким чином, процедури можуть отримувати і повертати значення не тільки через параметри-результати, але й через глобальні змінні (тому у процедурі може і не бути параметрів).

**3. Структура, опис підпрограми-функції. Вхідні і вихідні параметри**

Ми вже з’ясували, як описується підпрограма-процедури і як вона викликається з основної програми.

Підпрограма-процедура відрізняється від підпрограми-функції тим, що вона, виконуючи потрібні дії, не повертає ніякого значення.

Тому можна сказати, що процедура – частковий випадок функції.

Розглянемо ***загальний опис функції:***

***function <ім’я> (<список формальних параметрів>):<тип>;***

***<розділ описів і оголошень функції>;***

***begin***

***<розділ команд функції>***

***end;***

***В розділі команд функції обов’язково повинен бути оператор присвоєння імені функції значення результату: <ім’я функції >:=результат;***

Результат функції повертається в основну програму через її ім’я із зазначенням фактичних параметрів. ***Під час виклику функції тип не вказується:***

***<ім’я > (<список фактичних параметрів >);***

**Закріплення знань:**

**Запитання:**

1. Що таке підпрограма? Які типи підпрограм Ви знаєте?

2. Чим відрізняються процедури від функцій?

3. Формати опису процедур і функцій.

4. Що таке локальні та глобальні параметри? Яка між ними відмінність?

5. Які параметри називаються фактичними, а які формальними?

6. Змінні яких типів можуть використовуватись в якості параметрів процедур та функцій?

7. Чи можуть існувати підпрограми без формальних параметрів?

8. Чим відрізняються параметри-значення від параметрів-змінних?

9. Що можна ставити у відповідність формальним параметрам-значенням, а що параметрам-змінним?

**Домашнє завдання:**

1. Вивчити конспект.
2. Дати письмову відповідь на запитання.