**УРОК №25.**

**ТЕМА:** ***Стандартні підпрограми та підпрограми користувача.***

**Тема:** **Стандартні підпрограми та підпрограми користувача.**

**Мета:** Актуалізувати знання учнів про підпрограми та їх аргументи, локальні та глобальні змінні, формальні та фактичні параметри підпрограм. Подати новий навчальний матеріал про стандартні підпрограми та підпрограми користувача в Delphi. Закріпити даний матеріал під час виконання практичних завдань.

**Тип уроку:** Комбінований урок.

**План уроку:**

Актуалізація опорних знань учнів.

Вивчення нового навчального матеріалу.

1. Процедури користувача в Delphi.
2. Функції користувача в Delphi.
3. Стандартні процедури та функції в Delphi.

Практичні завдання.

Домашнє завдання.

**Актуалізація опорних знань:**

**1.** Для чого використовують підпрограми? Які види підпрограм виділяють?

**2.** Які формати опису процедур і функцій?

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

**4.** Які параметри називаються фактичними, а які формальними?

**Вивчення нового навчального матеріалу:**

**1. Процедури користувача в Delphi.**

Раніше ви створювали і використовували процедури, які були пов’язані з певними подіями і були методами певних об’єктів.

Заготовки цих процедур середовище розробки створювало автоматично, а ви доповнювали ці заготовки командами і рядком varоголошення змінних.

Під час створення програм можна включати до проекту процедури, не пов’язані з подіями. Такі процедури називаються процедурами користувача. Їх найчастіше використовують у випадках, коли потрібно кілька разів при різних значеннях змінних виконувати одні й ті самі операції. Або коли задачу можна поділити на окремі підзадачі, кожна з яких розв’язується за окремим алгоритмом, тобто при використанні методу покрокової деталізації.

***Процедура користувача, як й інші процедури, – це відокремлена частина програми, яка має своє ім’я та яка може бути викликана для виконання в будь-якій іншій частині програми.***

Процедури користувача можуть бути:

* без аргументів і без результатів;
* з аргументами і без результатів;
* без аргументів і з результатами;
* з аргументами і з результатами.

***Аргументи процедури* – це змінні, які отримують свої значення в команді виклику процедури перед початком виконання команд процедури.**

***Результати процедури* – це змінні, які передають свої значення з процедури певним змінним у команді виклику процедури.**

Найчастіше використовують процедури останнього типу. Загальний вигляд цих процедур у **Delphi** такий:

**procedure <ім’я> (<списки аргументів з їхніми типами>; var <списки результатів з їхніми типами>);**

**var (<списки локальних змінних процедури з їхніми типами>);**

**begin**

**<команди>**

**end;**

Загальний вигляд команди виклику процедури з аргументами і результатами такий:

**<ім’я> (<списки виразів, значення яких передаються аргументам процедури>, <списки змінних, яким передають свої значення результати процедури>);**

*Зверніть вашу увагу:* у команді виклику процедури порядок виразів, значення яких передаються аргументам процедури, і змінних, яким передають свої значення результати процедури, повинен строго відповідати порядку запису аргументів і результатів процедури в рядку її заголовка.

**Змінні, які вказані в заголовку процедури, називаються *формальними параметрами процедури*.**

**Змінні, які вказуються в команді виклику процедури, називаються *фактичними параметрами процедури*.**

**2.** **Функції користувача в Delphi.**

Як зазначалося вище, процедура може не мати результатів, а може мати один або більше результатів. Якщо процедура має один результат, то зручніше її оформити і викликати як **функцію користувача**.

***Функція користувача, як і процедура користувача – це відокремлена частина програми, яка маєсвоє ім’я та яка може бути викликана для виконання в будь-якій іншій частині програми.***

Відрізняється функція від процедури таким:

1. Рядок заголовка функції виглядає так:

**function <ім’я> (<списки аргументів з їхніми типами>): <тип результату >;**

2. Значення, яке функція повинна повернути (передати) команді, яка її викликає, у тексті самої функції має бути присвоєно змінній, ім’я якої збігається з іменем функції.

3. Функція може бути викликана або в правій частині команди присвоювання, або в команді виведення.

**3. Стандартні процедури та функції в Delphi.**

У мові програмування **Delphi** можна використовувати великий набір стандартних процедур і функцій, призначених для виконання різноманітних операцій над числами, текстами, елементами керування, графічними об’єктами, файлами та ін.

Деякі з них ви вже використовували, наприклад функцію **StrToInt** для переведення тексту в ціле число, функцію **FloatToStr** для переведення дійсного числа в текст, функцію **sqrt** для обчислення арифметичного квадратного кореня з дійсного числа.

Розглянемо ще кілька стандартних математичних функцій та їхнє призначення (табл.).

*Зверніть вашу увагу*:

1. При виклику функції без аргументів потрібно вказувати лише її ім’я, наприклад **х :=** **Pi**.

2. Перед першим використанням функції **Random** потрібно включити до тексту проекту команду виклику процедури **Randomize** без аргументів і без результатів, яка запустить генератор випадкових чисел.

3. Для використання функції **Power** потрібно в рядок **Uses** текступроекту додати текст **Math**. Після цього до проекту підключиться додаткова бібліотека стандартних процедур та функцій і стане можливим використання ще цілого ряду математичних та інших процедур і функцій, з якими ви можете ознайомитися в **Довідці**.

**Практичні завдання:**

**1.** Запишіть заголовок процедури з іменем ***stepin***, аргументами якої є змінна *a* дійсного типу і змінна *b* цілого типу, а результатом – змінна *с* дійсного типу, і приклад команди її виклику.

**2*.*** Запишіть процедуру з іменем ***square***, яка виконує обчислення площі трикутника за відомими довжинами його сторін, значення яких передаються до процедури як аргументи. Для обчислення площі використайте формулу Герона.

**3.** Запишіть заголовок функції з іменем ***length***, аргументами якої є чотири змінні цілого типу, а результат має дійсний тип, і приклад команди її виклику.

**4.** Запишіть функцію, яка виконує обчислення довжини відрізка за відомими координатами його кінців. Запишіть команду виклику цієї функції для обчислення довжини відрізка *а* з координатами кінців (0; 0) та (20; 30).

**5.** Створіть проект, у якому можна знайти максимальне із чотирьох заданих різних цілих чисел. Використайте функцію для знаходження більшого з двох заданих цілих чисел. Створіть у власній папці папку **Проект 25** і збережіть у ній проект.

**Домашнє завдання:**

1. Вивчити конспект.
2. Опрацювати матеріал підручника на ст. 95-100. (Й.Я. Ривкінд, Т.І. Лисенко, Л.А. Чернікова, В.В. Шакотько).
3. Завершити виконання практичних завдань.