

Алгоритми з повтореннями.

Оператори циклу мовою програмування.

Мета.

Навчальна. Ознайомити учнів з командою повторення та її видами, способами подання команди повторення мовою програмування, з операторами циклу (з передумовою та післяумовою, з параметром).

Розвиваюча. Розвивати логічне мислення, навички створення програм, самостійність, вміння застосовувати набуті знання до практичних завдань.

Виховна. Виховувати наполегливість, естетичність у оформленні, грамотно висловлювати свої думки.

Тип уроку. Засвоєння нових знань і навичок.

Матеріали для роботи з учнями:**План**

Введення в тему.

Логічні вирази та логічні операції, таблиці істинності.

Практичні завдання.

Типові запитання до уроку.

Домашнє завдання.

Пам'ятка для учня!

Пригадайте правила техніки безпеки при роботі з ПК.

Через кожні 15 хв. виконуйте вправи для очей та для зняття м'язової втоми.

Хід уроку

1. Введення в тему.

Кількість годин відведених на вивчення теми.

Основні навички, якими повинен оволодіти учень по закінченню вивчення теми.

[Інструктаж з ТБ при роботі з ПК та в комп'ютерному класі.](#)

2. Мотивація.

Більшість процесів у житті неодноразово повторюються (пори року, день — ніч, розпорядок дня, графік руху поїздів), тому немає потреби в програмах десятки разів записувати одні й ті самі команди, а достатньо вказати команди лише один раз і здійснити повторення за допомогою операторів циклу. Розглянемо, як це роблять мовою програмування.

3. Алгоритми з повтореннями. Оператори циклу мовою програмування.

Циклом у програмуванні називається повторення одних і тих самих дій. Цикл закінчується за якої-небудь умови.

Цикл - це алгоритмічна структура, за допомогою якої реалізується багаторазове повторення блоків операторів.

Вказівка повторення з передумовою (Цикл—while)

Вказівка повторення з передумовою — while *призначена* для організації багатократного виконання групи вказівок (тіло циклу) до тих пір, поки залишається істинною умова виконання циклу.

Значення службового слова *while* — *поки*

Синтаксис оператора *while*:

while <умова> do <оператор>;

де <умова> - вираз логічного типу;

<оператор> - простий або складений оператор Паскаля.

Простий або складений оператор, який виконується при кожному повторенні циклу, називається *тілом циклу*.

Вказівка повторення з передумовою записується у вигляді:

```
while <умова> do
    begin
        <вказівка1>;
        <вказівка2>;
        .....
        <вказівка N>
    end;
```

Наприклад,

```
while I<4 do
    begin
        S:=S+I;
        I:=I+1
    end;
```

Виконується ця вказівка таким чином. Обчислюється значення виразу-умови. Якщо умова істинна (приймає значення TRUE), то виконується група вказівок (тіло

циклу). Виконання програми знову повертається до перевірки умови циклу. Якщо умова хибна (приймає значення FALSE), то виконується вказівка, яка знаходиться після вказівки повторення з передумовою.



Примітка.

1. Якщо умова у вказівці повторення хибна при першій же перевірці, то тіло циклу не виконується жодного разу. Умова вказівки є умовою входження в цикл.
2. Умову вказівки повторення з передумовою необхідно підбирати таким чином, щоб в деякий момент вона змінила своє значення з TRUE на FALSE, інакше виникне ситуація "зациклювання" програми.
3. Програмісту необхідно самому потурбуватись про необхідність збільшення (зменшення) лічильника циклу на 1. Це можна зробити таким чином $i:=i+1$.
4. Входження в тіло вказівки повторення можливе тільки через заголовок цієї вказівки.

Приклад використання вказівки повторення з передумовою

Задача.1 Знайти суму всіх натуральних чисел від i до N .

```

1  Program SUMA1;
.  var i,N,S:integer;
.  begin
4    writeln('Vvedit kilkist naturalnyh chisel:');
5    write('N=');
.    readln(N);
.    i:=1;
.    S:=0;
.    while i<=N do
10   begin
.     S:=S+i;
.     i:=i+1;
.     end;
.     writeln('S=',S) ;
15    readln
.    end.
17

```

Вказівка повторення з післяумовою (Цикл-repeat-until)

Вказівка повторення з післяумовою призначена для організації багатократного виконання групи вказівок (тіло циклу) до тих пір, поки умова виконання циклу не стане істинною.

Синтаксис оператора

repeat:

**repeat <оператори> until
<умова>;**

де <умова> - вираз логічного типу;

<оператор> - послідовність операторів Паскаля, яка складає тіло циклу.

Вказівка повторення з передумовою
записується у вигляді:

repeat

<вказівка1>;

<вказівка2 >;

.....;

<вказівка N>
until <умова>;

Наприклад.

```
repeat  
  S:=S+I;  
  I:=I+1  
until I>=4;
```

Службові слова вказівки

означають:

repeat — повторювати, **until** — поки.

Вказівка повторення з післяумовою виконується таким чином. Спочатку виконуються вказівки, які входять в тіло циклу. Потім обчислюється значення умови-виразу. Якщо умова хибна (тобто має значення FALSE), то знову виконуються вказівки тіла циклу. Якщо умова стає істинною (TRUE), то виконання циклу припиняється і виконується наступна вказівка програми.



Примітка.

1. Тіло циклу, яке складається з групи вказівок, не потрібно брати в дужки (begin — end), оскільки службові слова repeat — until відіграють роль дужок.
2. Тіло циклу у вказівці повторення з післяумовою виконується щонайменше один раз.

3. Умова вказівки є умовою виходу з циклу.
4. Входження в тіло вказівки повторенням можливе тільки через заголовок цієї вказівки.

Приклад використання вказівки повторення з післяумовою

Задача.2 Знайти суму всіх натуральних чисел від 1 до N.

 Редактор тексту

*project1.lpr

```
1  Program SUMA2;  
  .  var i,N,S:integer;  
  .  begin  
  .    writeln('Введіть кількість натуральних чисел:');  
5   write('N=');  
  .   readln(N);  
  .   i:=1;  
  .   S:=0;  
  .   repeat  
10  S:=S+i;  
  .   i:=i+1;  
  .   until i>N;  
  .   writeln('S=',S);  
  .   end.  
15  
16
```

В мові програмування Turbo Pascal логічна функція KeyPressed контролює натискання клавіш. Якщо жодна з клавіш не натиснута, ця функція приймає значення False, інакше - True.

Конструкція **Repeat Until KeyPressed** використовується для організації паузи в процесі виконання програми (до натискання будь-якої клавіші).

Вказівка повторення з параметром (FOR — TO — DO).

Вказівка повторення з параметром призначена для організації багатократного виконання тіла циклу для значень параметра циклу з деякої впорядкованої дискретної множини.

Вказівку повторення з параметром використовують у тих випадках, коли наперед можна визначити кількість повторень виконання вказівок циклу.

Вказівка повторення з параметром записується у вигляді:

1. **for <параметр:=вираз1> to <вираз2> do**

```

begin
  <вказівка1>;
  <вказівка 2>;
  .....
  <вказівка N>;
end;

```

2. for <параметр:=вираз1> downto <вираз2> do

```

begin
  <вказівка1>;
  <вказівка2>;
  .....
  <вказівка N>;
end;

```

Параметр вказівки (лічильник, індекс) — це змінна скалярного (цілого, символьного, логічного або перелічувального) типу, крім дійсного.

Вираз 1 (початковий параметр вказівки) — це деяка константа такого ж самого типу, що й параметр вказівки.

Вираз 2 (кінцевий параметр вказівки) — це деяка константа такого ж самого типу, що й параметр вказівки.

Якщо в тілі вказівки повторення з параметром необхідно виконати тільки одну вказівку, то складеної вказівки **begin .. end** використовувати не потрібно.

```

for <параметр:=вираз1> to <вираз2> do
  <вказівка1>;

```

Вказівка повторення з параметром виконується таким чином:

1. Вказівка For — to — do.

Наприклад.

```

for i:=K to N do
begin
  <вказівка1>;
  <вказівка2>;
  .....
  <вказівка N>;
end;

```

Параметру циклу *i* присвоюється початкове значення *K*. Він порівнюється з кінцевим значенням *N*. Якщо $K \leq N$, то виконується тіло вказівки повторення. Значення *K* автоматично збільшується на 1 (тобто стає наступним елементом) і знову

порівнюється зі значенням N. Якщо під час перевірки отримаємо, що $K > N$, то виконання вказівки повторення припиняється і виконується наступна після неї вказівка програми. Якщо під час першого порівняння K і N виявиться, що $K > N$, то тіло вказівки не виконується жодного разу.

2. Вказівка For — downto — do.

Наприклад.

```
for i:=K downto N do  
    begin  
        <вказівка1>;  
        <вказівка2>;  
        .....  
        <вказівка N>;  
    end;
```

Параметру циклу i присвоюється початкове значення K. Він порівнюється з кінцевим значенням N. Якщо $K \geq N$, то виконується тіло вказівки повторення. Значення K автоматично зменшується на 1 (тобто стає попереднім елементом) і знову порівнюється зі значенням N. Якщо під час перевірки отримаємо, що $K < N$, то виконання вказівки повторення припиняється і виконується наступна після неї вказівка програми. Якщо під час першого порівняння K і N виявиться, що $K < N$, то тіло вказівки не виконується жодного разу.

Примітка.

1. Вказівку повторення з параметром зручно використовувати в тих випадках, коли попередньо можна визначити кількість повторень.
2. Параметр циклу не повинен змінювати свого значення всередині тіла вказівки повторення, хоча його можна використовувати у будь-яких виразах вказівок тіла циклу.
3. Параметр циклу повинен бути описаним у розділі змінних.
4. Значення параметра циклу після закінчення виконання вказівки повторення володіє кінцевим значенням, тому перед повторним його використанням йому необхідно знову присвоїти нове початкове значення.

Приклад використання вказівки повторення з параметром

Задача.3 Знайти суму всіх натуральних чисел від 1 до N.

```
1  Program SUMA3;  
  . var i,N,S: integer;  
  . begin  
  .   writeln('Введіть кількість натуральних чисел:');  
5  write('N=');  
  . readln(n);  
  . S:=0;  
  . for i:=1 to N do  
  .   S:=S+i;  
10  writeln('S=',S);  
  . readln;  
  . end.  
13
```

4. Практичні завдання.

1. Знайти значення: $1 \cdot 3 \cdot 5 \cdot \dots \cdot 101$.
2. Визначити:
 - 1) $n!$;
 - 2) 2^n ;

5. Підсумки уроку.

1. Поясніть синтаксис і правила виконання оператора циклу з передумовою.
2. Поясніть синтаксис і правила виконання оператора циклу з післяумовою.
3. Поясніть структуру і правила виконання оператора циклу з параметром.

6. Домашнє завдання.

1. Конспект.
2. Визначити: $2+4+6+\dots+2n$.
3. [Тест](#).