



Й.Я. Ривкінд, Т.І. Лисенко,
Л.А. Чернікова, В.В. Шакотько

ІНФОРМАТИКА

8



**ЦИКЛ З ПЕРЕДУМОВОЮ.
СПІВВІДНОШЕННЯ ТИПІВ
ДАНИХ ТА ЕЛЕМЕНТІВ
ДЛЯ ВВЕДЕННЯ ДАНИХ,
ЗЧИТУВАННЯ ДАНИХ З
ЕЛЕМЕНТІВ ВВЕДЕННЯ**



Visual C++



python

Programming Language

Free Pascal



Урок 55

РОЗДІЛ 6 § 6.7



1. **Що таке цикл в алгоритмі?**
2. **Який вигляд має команда циклу з лічильником в Object Pascal?**
3. **Як виконується команда циклу з лічильником в Object Pascal?**
4. **Який вигляд має блок-схема циклу з лічильником? Поясніть виконання фрагмента алгоритму.**
5. **Чи можуть команди тіла циклу з лічильником виконуватися тільки один раз? Поясніть свою відповідь. Наведіть приклад.**





Ви вже знаєте, що цикл з передумовою як фрагмент алгоритму починається з команди перевірки умови й результатом виконання цієї команди може бути:

Істина (Так, true)

Хиба (Ні, false)

І залежно від результату виконання цієї команди – виконуватимуться:

Команди тіла циклу

або

Команда алгоритму, наступна за циклом



У загальному випадку у цій команді визначається значення певного логічного виразу, яке може бути:

True

або

False





Загальний вигляд блок-схеми **циклу з передумовою** такий:



Команда циклу з передумовою в Object Pascal



Команда **циклу з передумовою** в мові програмування *Object Pascal* має такий вигляд:

```
While <логічний вираз>  
Then begin  
<команди тіла циклу>  
end;
```

(англ. **while** - **поки**).

Команда циклу з передумовою в Object Pascal



Виконання цієї команди відбувається так: обчислюється значення логічного виразу; якщо це значення **true**, то виконуються команди тіла циклу, після чого знову обчислюється значення логічного виразу, і якщо це значення знову **true**, то знову виконуються команди тіла циклу, після чого знову обчислюється значення логічного виразу;

якщо значення логічного виразу дорівнює **false**, то команди тіла циклу не виконуються, а виконується команда, наступна за циклом.



Команда циклу з передумовою в Object Pascal



Звертаємо вашу увагу:

- ✓ якщо в тілі циклу лише одна команда, то операторні дужки **begin** і **end** можна не ставити;
- ✓ серед команд тіла циклу можуть бути й лінійні фрагменти, і розгалуження, й інші цикли.



Команда циклу з передумовою в Object Pascal



Розглянемо задачу, алгоритм розв'язування якої містить цикл з передумовою.

Задача. Скільки потрібно взяти доданків, перший з яких дорівнює заданому дійсному числу, а кожний наступний на 3 більший від попереднього, щоб їхня сума перевищила 100?

Команда циклу з передумовою в Object Pascal



Даними цієї задачі є перший доданок. Вводимо його в поле.

Перед початком циклу присвоїмо:

Змінній *s*

яку використовуємо для збереження суми, значення першого доданка

Змінній *n*

яку використовуємо для збереження кількості взятих доданків, присвоїмо значення 1, бо в сумі вже враховано один (перший) доданок.

Команда циклу з передумовою в Object Pascal



*До суми потрібно додавати наступні доданки, поки ця сума буде меншою, або дорівнюватиме **100**. Тому логічний вираз у заголовку циклу матиме вигляд*

$$s \leq 100$$

У тілі циклу шукатимемо наступний доданок, додаватимемо його до поточного значення суми і збільшуватимемо кількість доданих доданків на 1.

Команда циклу з передумовою в Object Pascal



Відповідний фрагмент проекту матиме такий вигляд:

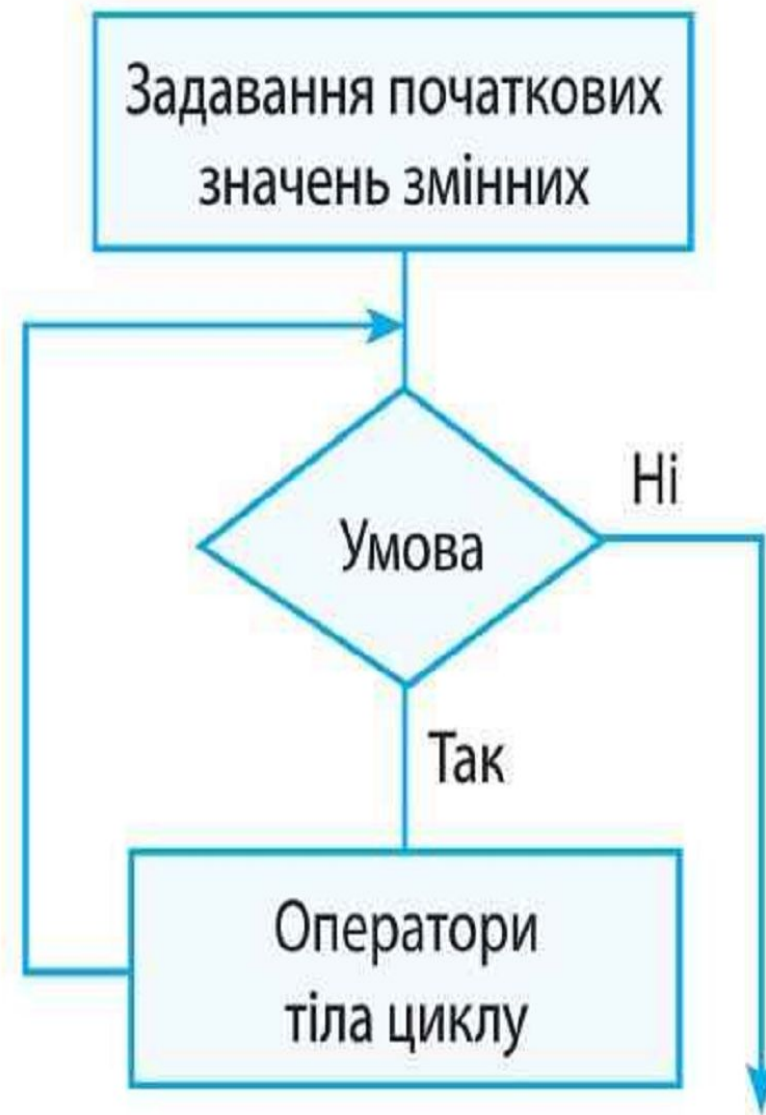
```
var a, s: real; n: integer;  
begin  
  a := StrToFloat (Edit1.Text);  
  s := a; n := 1;  
  while s <= 100 do  
  begin  
    a := a+3;  s := s+a;  
    n := n+1;  
  end;  
  Label1.Caption := IntToStr(n);  
end;
```

Оператор циклу з передумовою While



Необхідно задавати початкові значення змінних, використуваних у циклі.

Якщо при першій перевірці умова виявиться хибною, тіло циклу *While* не виконається жодного разу.



Оператор циклу з передумовою While



Виконати цикл із передумовою при різних початкових значеннях змінної:

While X <= 10 do X := X+1;

<i>Початкове значення X</i>	<i>Значення X після виконання циклу</i>	<i>Кількість повторень</i>
<i>7</i>	<i>11</i>	<i>4</i>
<i>10</i>	<i>11</i>	<i>1</i>
<i>15</i>	<i>15</i>	<i>0</i>

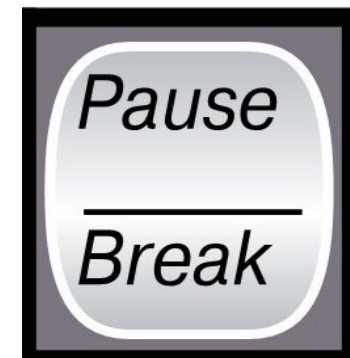
Оператор циклу з передумовою While



Логіка цього циклу змушує програму працювати вічно (нескінченний цикл).

```
Num := 0;  
While Num < 20 do  
Label1.Caption := IntToStr(Num)
```

Для переривання нескінченного циклу треба натиснути сполучення клавіш **Ctrl + Break.**





Поки A менше за 100, збільшувати значення A на одиницю і виводити значення змінної у Label.

Для призупинення роботи програми на 100 мілісекунд використано процедуру *Sleep*.

```
var A: Integer;
```

```
begin
```

```
  A := 1; // Початкове значення A
```

```
  While A < 100 do
```

```
  begin
```

```
    A := A + 1;
```

```
    Label1.Caption := IntToStr(A);
```

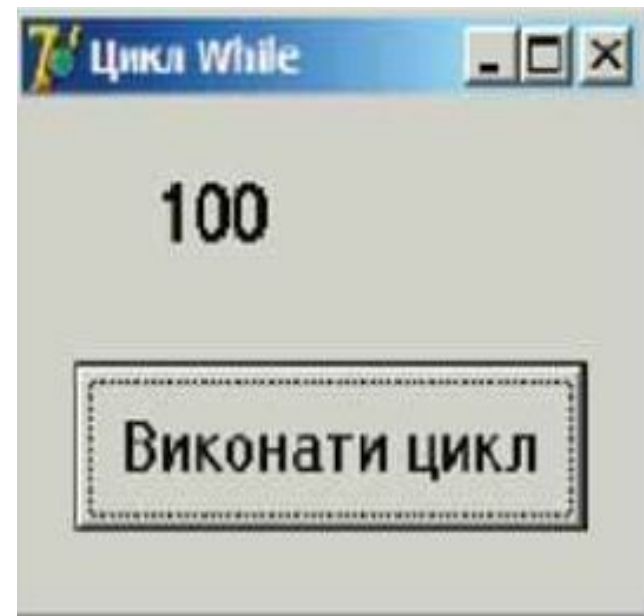
```
    Application.ProcessMessages; //
```

```
  обробка повідомлень
```

```
    Sleep(100); // зупинка на 0,1 с
```

```
  end;
```

```
end;
```



Алгоритм обчислення n -го члена послідовності натуральних чисел і суми n членів складається з таких дій:

задаються початкові значення A , — першого члена послідовності,
 n — кількості членів, які потрібно обчислити;

задаються початкові значення лічильника членів послідовності ($i := 0$) і суми ($S := 0$);

поки номер i -то члена послідовності, який обчислено, не досягне значення n — заданої кількості членів, повторюються дії: номер поточного доданка збільшується на 1; обчислюється значення наступного доданка A , обчислене значення A додається до суми S .



Алгоритм обчислення суми n членів числової послідовності



**Знайти суму 20 елементів послідовності чисел
2,5; 3,0; 3,5; 4,0...**

// Задаються початкові значення:

$S := 0;$ *// суми S*

$i := 0;$ *// лічильника доданків i*

$A := 2.5;$ *// доданка A*

While $i < 20$ do begin

$S := S + A;$ *// додається до суми S*

$A := A + 0.5;$ *// обчислюється наступне значення доданка*

A

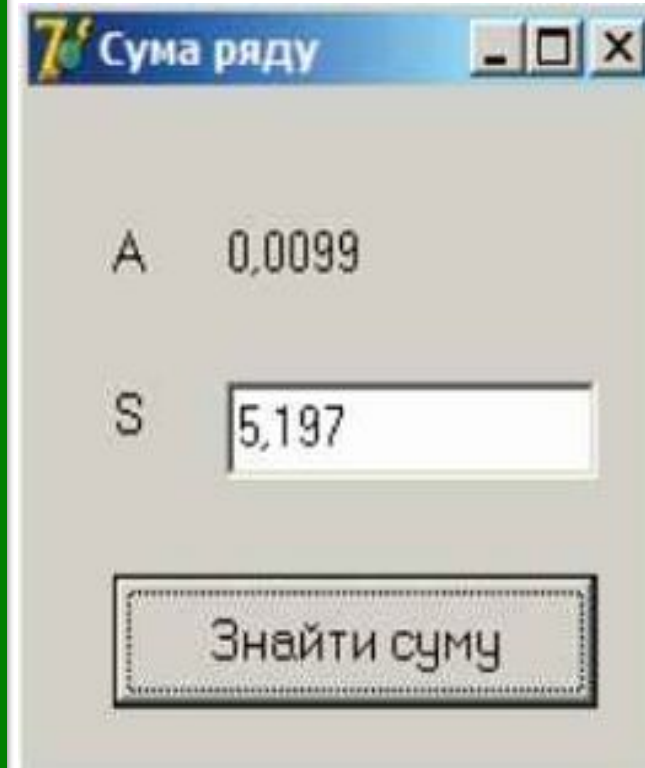
$i := i + 1$

end;

Знайти суму всіх елементів послідовності $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \dots$, значення яких не менше за 0,01.

Під час складання програми будемо вважати, що точність досягнуто, якщо черговий доданок менший за значення точності.

```
var i: Integer; A, S: Double;  
begin  
    // Задаються:  
    S:=1; // початкове значення суми S  
    A:=1; // значення першого доданка  
A  
    i:=1; // номер першого доданка  
    While A>0.01 do begin  
        i := i + 1;  
        A := 1/i; // Обчислення доданка A  
        S := S + A;  
        Label2.Caption := FormatFloat('0.####',  
A);  
    end;  
    Edit1.Text := FormatFloat('0.###', S);  
end;
```



Сума ряду

A 0,0099

S 5,197

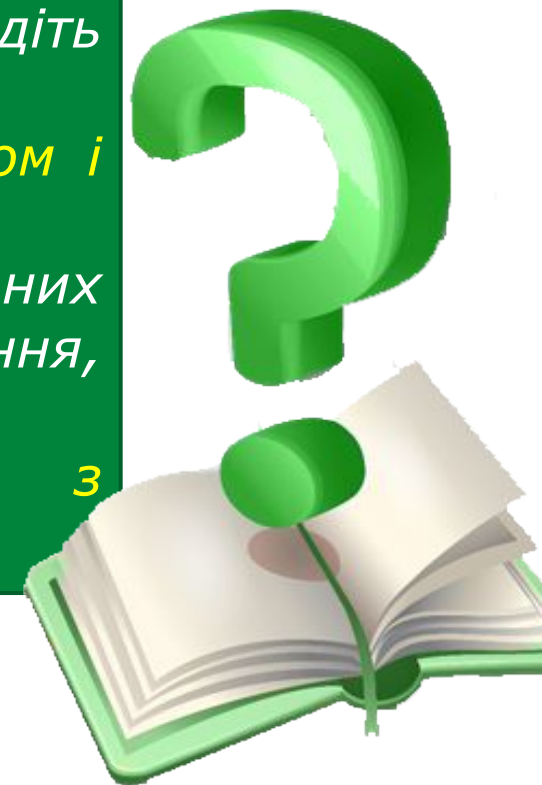
Знайти суму



Передумова



1. Який вигляд має блок-схема циклу з передумовою? Поясніть виконання цього циклу.
2. Чи можуть команди тіла циклу з передумовою не виконуватись жодного разу? Поясніть свою відповідь. Наведіть приклади.
3. Чи може виконання циклу з передумовою ніколи не закінчитися? Поясніть свою відповідь. Наведіть приклади.
4. Чим відрізняються між собою цикл з лічильником і цикл з передумовою?
5. У чому полягають відмінності у виконанні основних алгоритмічних структур: слідування, розгалуження, цикл з лічильником і цикл з передумовою?
6. Який загальний вигляд команди циклу з передумовою в *Object Pascal*?



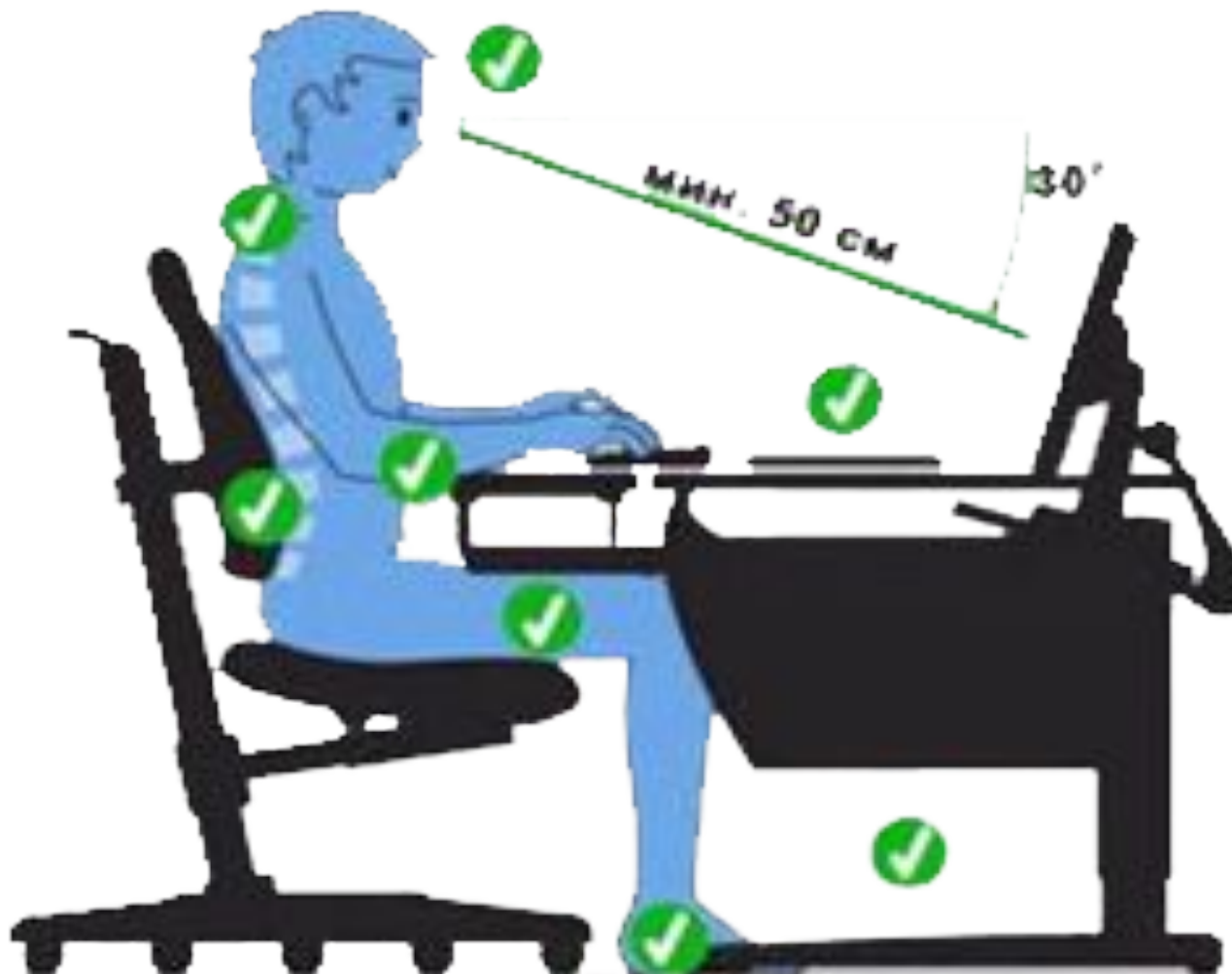


Фізкультхвилинка





Працюємо за комп'ютером





Й.Я. Ривкінд, Т.І. Лисенко,
Л.А. Чернікова, В.В. Шакоцько

ІНФОРМАТИКА

8



**ДЯКУЮ
ЗА УВАГУ!!!**



Free Pascal



python

Programming Language



Visual C++

Урок 43

РОЗДІЛ 6 § 6.7