

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»
ФАКУЛЬТЕТ МАТЕМАТИКИ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
Кафедра системного аналізу та теорії оптимізації

А.Ю.Брила, М.М. Ломага, П.П. Антосяк

РЕАЛІЗАЦІЯ ЦИКЛІЧНИХ АЛГОРИТМІВ У С#

Методичні матеріали з організації самостійної роботи студентів з дисципліни
«Програмування»

Ужгород 2023

Реалізація циклічних алгоритмів у С#. (Методичні матеріали з організації самостійної роботи студентів з дисципліни «Програмування»). /Укладачі: А.Ю. Брила, М.М. Ломага, П.П. Антосяк. – Ужгород, 2023. – 15 с.

Навчальний матеріал методичних вказівок призначений для аудиторної і самостійної підготовки студентів при вивченні дисципліни «Програмування».

Основна мета самостійної роботи студента – закріплення теоретичних відомостей, які викладаються на лекціях, та вміння їх застосувати, розв’язуючи задачі, що виникають на практиці. Завдання розроблених методичних матеріалів полягає в чіткій, цілеспрямованій допомозі студентам в організації самостійної підготовки до практичних занять з дисципліни «Програмування». Розроблена система тестових питань по темі «Оператори циклів. Реалізація циклічних алгоритмів» дає змогу поглибити знання та уміння їх застосовувати при розробці алгоритмів.

Методичні вказівки призначені для студентів різних напрямків підготовки.

Рецензенти:

к.ф.-м.н., доц. Погоріляк О.О.,

к.т.н., доц. Андрашко Ю.В.

Рекомендовано до друку:

Кафедрою системного аналізу та теорії оптимізації (Протокол №10 від 18 травня 2023 року);

Науково-методичною комісією факультету математики та цифрових технологій, (Протокол № 9 від 23 травня 2023 року);

Вченою радою факультету математики та цифрових технологій ДВНЗ “Ужгородський національний університет”, (Протокол №9 від 25 травня 2023 року).

РЕАЛІЗАЦІЯ ЦИКЛІЧНИХ АЛГОРИТМІВ

Оператори циклу

Часто постає потреба виконати один і той самий оператор декілька разів. Для цього застосовують *оператори циклів*. Цикл складається із *заголовка* і *тіла*. У заголовку циклу зазначається умова завершення циклу, а тіло циклу являє собою оператор, який потрібно виконати декілька разів. Кожне виконання оператора тіла циклу називається його *ітерацією*.

Оператор циклу while

Оператор while циклічно виконує своє тіло до тих пір, поки умова виконується (логічний вираз приймає значення true).

Програмна структура	Аналог на мові блок-схем	Приклад. Знайти суму перших n натуральних чисел.
<pre>while (<умова>) <оператор>;</pre>		<pre>int sum=0; int i=1; while(i<=n) sum=sum+i++;</pre>

Якщо тіло циклу складається з більше ніж одного оператора, то необхідно використати складений оператор.

Програмна структура	Аналог на мові блок-схем	Приклад. Знайти суму і добуток перших n натуральних чисел.
<pre>while (<умова>) { <оператор1>; <операторN>; }</pre>		<pre>int sum=0; int mult=1; int i=1; while(i<=n) { sum=sum+i; mult=mult*(i++); }</pre>

Оператор циклу do-while

Оператор циклу do-while відрізняється від оператора while тим, що перевірка умови виконується не до, а після виконання інструкції (оператора). У операторі циклу do-while тіло циклу виконається принаймні один раз.

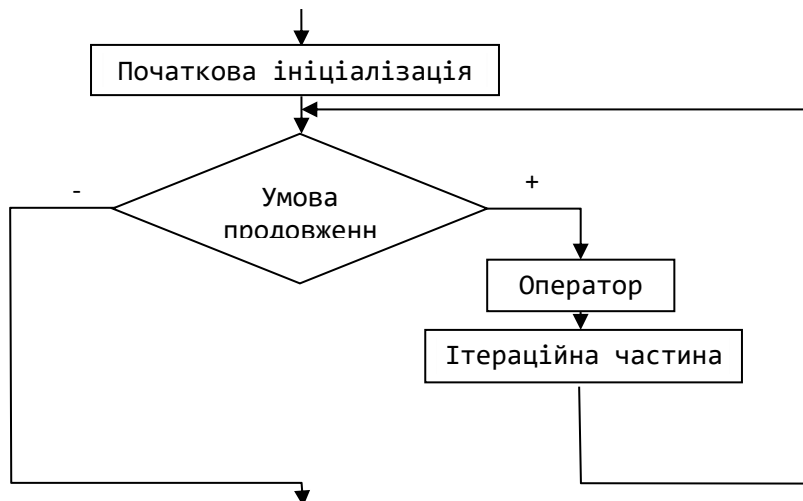
Програмна структура	Аналог на мові блок-схем	Приклад. Знайти суму перших n натуральних чисел.
<pre>do { <оператор>; } while (<умова>)</pre>		<pre>int sum=0; int i=1; do { sum=sum+i++; } while(i<=n)</pre>

Оператор циклу for

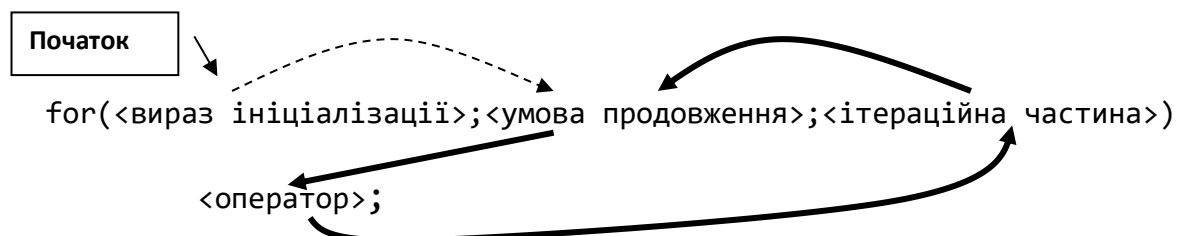
Загальний вигляд:

```
for (<вираз ініціалізації>;<умова продовження>;<ітераційна частина>)
    <оператор>;
```

Аналог на мові блок-схем:



Схематичне зображення виконання оператора:



Оператор `for` працює у відповідності до наступного алгоритму:

1. Обчислюється вираз ініціалізації. У цій частині допустима ініціалізація декількох лічильників циклу.
2. Перевіряється умова продовження. Якщо умова невірна то робота циклу завершується і передається управління наступному оператору.
3. Якщо умова істинна, виконується тіло даного оператора.
4. Виконується приріст одного або декількох лічильників циклу (або виконується довільна інша операція).
5. Здійснюється перехід до кроку 2.

Приклад. Знайти суму перших n натуральних чисел.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int n,sum=0;
            Console.WriteLine("n=");
            n=int.Parse(Console.ReadLine());
            for (int i = 1; i <= n; i++)
                sum = sum + i;
            Console.ReadKey();
        }
    }
}
```

Проілюструємо випадок, коли частина ініціалізації та ітераційна частина містять декілька виразів, розділених комою.

Приклад. Обчислити значення виразу

$$\frac{1}{0,3} + \frac{2}{0,4} + \dots + \frac{n}{0,3 + 0,1 * (n - 1)}.$$

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
```

```

class Program
{
    static void Main(string[] args)
    {
        int n;
        Console.WriteLine("n=");
        n=int.Parse(Console.ReadLine());
        double d;
        int c;
        double sum=0;

        for (c = 1, d=0.3; c <= n; c++, d+=0.1)
            sum +=c/d;

        Console.WriteLine("sum={0:f3}", sum);
        Console.ReadKey();
    }
}

```

Кожна з частин оператора циклу може бути відсутньою, але розділові знаки “;” є обов’язковими. Так нескінчений цикл може бути задано так:

```

for ( ; ; )
    <оператор>;

```

Оператори break та continue

Оператор continue може бути використаний у будь-якому із циклів у випадку, коли немає потреби виконувати всі оператори у тілі циклу у поточній ітерації, а необхідно одразу перейти до наступної ітерації.

Приклад. Знайти добуток непарних натуральних чисел, що менші за K.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int K;
            Console.Write("K=");
            K=int.Parse(Console.ReadLine());
            int mult=1;

```

```

        for (int i = 1; i <= K; i++)
        {
            if ((i % 2) == 0) continue;
            mult *= i;
        }

        Console.ReadKey();
    }
}

```

Оператор `break` у циклах використовують для негайного завершення самого внутрішнього циклу, у тілі якого він знаходиться.

Приклад. Знайти найменше значення факторіалу натурального числа, що перевищує число `K`.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int K;
            Console.Write("K=");
            K = int.Parse(Console.ReadLine());
            int fakt = 1;
            for (int i = 1; ; i++)
            {
                fakt *= i;
                if (fakt > K) break;
            }
            Console.WriteLine("fakt={0}", fakt);
            Console.ReadKey();
        }
    }
}

```

Приклад. Вивести на екран усі натуральні двоцифрові числа, у яких друга цифра не перевищує першу.

При розв'язанні цього завдання використаємо оператор `break`, який здійснює вихід тільки із вкладеного циклу (робота зовнішнього циклу продовжується).

```

using System;

```

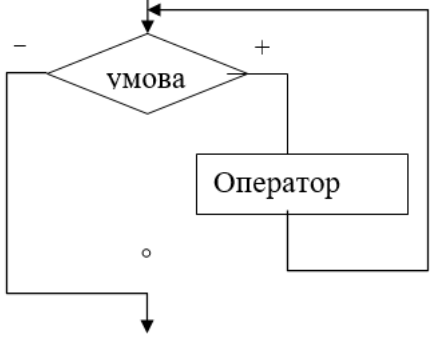
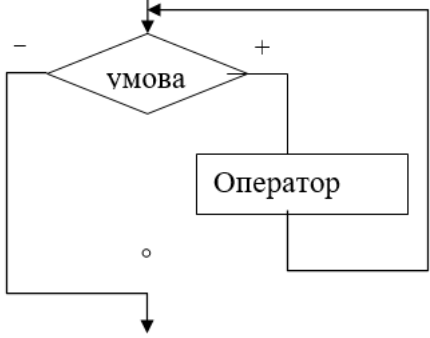
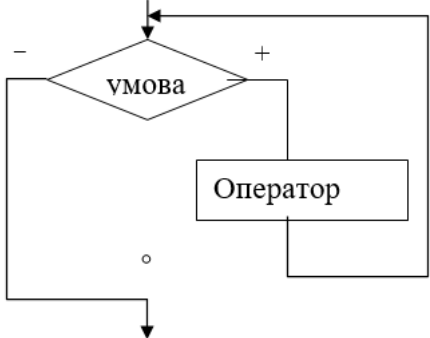
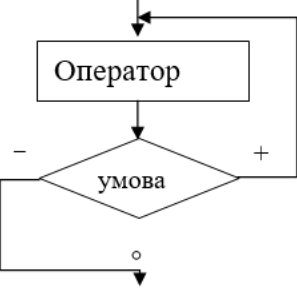
```
using System.Collections.Generic;
using System.Linq;
using System.Text;

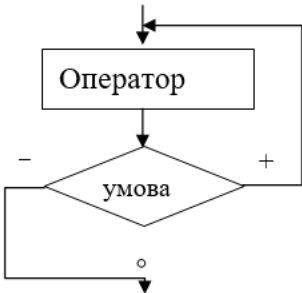
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            for (int i = 1; i <= 9; i++)
            {
                for (int j = 1; j <= 9; j++)
                {
                    if (j > i) break;
                    Console.WriteLine("{0}{1}", i, j);
                }
            }
            Console.ReadKey();
        }
    }
}
```

Питання для самоконтролю

1. У яких випадках застосовують оператори циклу?
2. Що таке тіло циклу?
3. Що називають ітерацією циклу?
4. Який загальний вигляд оператора циклу *while*?
5. Який загальний вигляд оператора циклу *do-while*?
6. Які відмінності у роботі операторів циклу *while* та *do-while*?
7. Який загальний вигляд оператора *for*?
8. За яким алгоритмом виконується оператор *for*?
9. Яке призначення оператора *break*?
10. Яке призначення оператора *continue*?

НАБІР ТЕСТІВ ДЛЯ САМОКОНТРОЛЮ

<p>1.</p>	<p>Який із операторів необхідно використати для реалізації вказаної частини блок-схеми?</p>  <p>1) оператор вибору; 2) умовний оператор; 3) оператор циклу; 4) немає правильної відповіді.</p>
<p>2.</p>	<p>Який із операторів необхідно використати для реалізації вказаної частини блок-схеми?</p>  <p>1) if-else; 2) switch; 3) while; 4) do-while; 4) немає правильної відповіді.</p>
<p>3.</p>	<p>Я називають цикл, що представлено на блок-схемі?</p>  <p>1) цикл з передумовою; 2) цикл з післяумовою; 3) скінченням; 4) немає правильної відповіді.</p>
<p>4.</p>	<p>Який із операторів необхідно використати для реалізації вказаної частини блок-схеми?</p>  <p>1) оператор вибору; 2) умовний оператор; 3) оператор циклу; 4) немає правильної відповіді.</p>

5.	<p>Який із операторів необхідно використати для реалізації вказаної частини блок-схеми?</p>  <p>1) if-else; 2) switch; 3) while; 4) do-while; 5) немає правильної відповіді.</p>
6.	<p>У яких із вказаних циклів можлива ситуація, коли тіло циклу не буде виконано жодного разу?</p> <p>1) while; 2) do-while; 3) for ; 4) немає правильної відповіді.</p>
7.	<p>У яких із вказаних циклів можлива ситуація, коли тіло циклу обов'язково буде виконано принаймні один раз?</p> <p>1) while; 2) do-while; 3) for ; 4) немає правильної відповіді.</p>
8.	<p>У яких із перерахованих циклів вказують умову завершення циклу?</p> <p>1) while; 2) do-while; 3) for ; 4) у жодному із перерахованих; 5) у кожному із перерахованих; 6) немає правильної відповіді.</p>
9.	<p>У яких із перерахованих циклів вказують умову продовження роботи циклу?</p> <p>1) while; 2) do-while; 3) for ; 4) у жодному із перерахованих; 5) у кожному із перерахованих; 6) немає правильної відповіді.</p>
10.	<p>Вкажіть чотири числа, які відповідають номерам складових частин циклу for у порядку їх слідування у цьому циклі.</p> <pre>for (<input style="width: 150px; border: 1px solid black;" type="text" value="перша складова"/> ; <input style="width: 150px; border: 1px solid black;" type="text" value="друга складова"/> ; <input style="width: 150px; border: 1px solid black;" type="text" value="третя складова"/>)</pre> <pre> <input style="width: 150px; border: 1px solid black;" type="text" value="четверта складова"/> ;</pre> <p>Перша складова:</p> <ol style="list-style-type: none"> 1) частина ініціалізації; 2) ітераційна частина; 3) оператор; 4) умова продовження; <p>Друга складова:</p> <ol style="list-style-type: none"> 5) оператор; 6) ітераційна частина; 7) частина ініціалізації; 8) умова продовження; <p>Третя складова:</p> <ol style="list-style-type: none"> 9) умова продовження; 10) оператор; 11) частина ініціалізації; 12) ітераційна частина; <p>Третя складова:</p> <ol style="list-style-type: none"> 13) умова продовження; 14) оператор; 15) частина ініціалізації;
11.	<p>Вкажіть два числа, які відповідають номерам складових частин циклу while у порядку їх слідування у цьому циклі.</p> <pre>while (<input style="width: 150px; border: 1px solid black;" type="text" value="перша складова"/>)</pre>

	<p><u>друга складова</u>;</p> <p>Перша складова:</p> <ol style="list-style-type: none"> 1) частина ініціалізації; 2) ітераційна частина; 3) оператор; 4) умова продовження; 5) умова завершення <p>Друга складова:</p> <ol style="list-style-type: none"> 6) оператор; 7) ітераційна частина; 8) частина ініціалізації; 9) умова продовження; 10) умова завершення
12.	<p>Вкажіть два числа, які відповідають номерам складових частин циклу do-while у порядку їх слідування у цьому циклі.</p> <pre>do <u>перша складова</u> while (<u>друга складова</u>);</pre> <p>Перша складова:</p> <ol style="list-style-type: none"> 1) частина ініціалізації; 2) ітераційна частина; 3) оператор; 4) умова продовження; 5) умова завершення <p>Друга складова:</p> <ol style="list-style-type: none"> 6) оператор; 7) ітераційна частина; 8) частина ініціалізації; 9) умова продовження; 10) умова завершення
13.	<p>Які із вказаних тверджень є істинними?</p> <ol style="list-style-type: none"> 1)цикл обов'язково повинен виконатися принаймні один раз; 2)цикл не можливо перервати; 3)цикл використовують у випадку, коли певну послідовність операторів треба повторювати певну кількість разів; 4)цикл використовують у випадку, коли певну послідовність операторів треба повторювати поки виконується вказана умова; 5)немає правильної відповіді.
14.	<p>Скільки разів може виконуватися вказаний оператор?</p> <pre>for (; ;) <оператор>;</pre> <p>1)жодного разу; 2) нескінчену кількість разів; 3) оператор містить помилку; 4) немає правильної відповіді.</p>
15.	<p>Скільки разів може виконуватися вказаний оператор?</p> <pre>while (true) <оператор>;</pre> <p>1)жодного разу; 2) нескінчену кількість разів; 3) оператор містить помилку; 4) немає правильної відповіді.</p>
16.	<p>Скільки разів буде виконано вказаний оператор?</p> <pre>while (false) <оператор>;</pre>

	1) жодного разу; 2) нескінчену кількість разів; 3) оператор містить помилку; 4) немає правильної відповіді.
17.	Скільки разів буде виконано вказаний оператор? <pre>do{ <оператор>; }while (false)</pre> 1) жодного разу; 2) нескінчену кількість разів; 3) оператор містить помилку; 4) немає правильної відповіді.
18.	Скільки разів буде виконано вказаний оператор? <pre>do{ <оператор>; }while (true)</pre> 1) жодного разу; 2) нескінчену кількість разів; 3) оператор містить помилку; 4) немає правильної відповіді.
19.	Скільки разів може виконуватися вказаний оператор? <pre>while (false) <оператор>;</pre> 1) жодного разу; 2) нескінчену кількість разів; 3) оператор містить помилку; 4) немає правильної відповіді.
20.	Скільки разів може виконуватися вказаний оператор? <pre>for (bool a=false ; ;) <оператор>;</pre> 1) жодного разу; 2) нескінчену кількість разів; 3) оператор містить помилку; 4) немає правильної відповіді.
21.	Скільки разів може виконуватися вказаний оператор? <pre>for (; true ;) <оператор>;</pre> 1) жодного разу; 2) нескінчену кількість разів; 3) оператор містить помилку; 4) немає правильної відповіді.
22.	Скільки разів може виконуватися вказаний оператор? <pre>for (bool a=false ; true ;) <оператор>;</pre> 1) жодного разу; 2) нескінчену кількість разів; 3) оператор містить помилку; 4) немає правильної відповіді.
23.	Який із операторів використовується для переривання циклу? 1) stop; 2) Stop; 3) continue; 4) немає правильної відповіді.
24.	При використанні якого із операторів одразу здійснюється перехід до наступної ітерації (всі інші оператори після вказаного оператора у поточній ітерації виконуватися не будуть)? 1) stop; 2) back; 3) next; 4) continue; 5) немає правильної відповіді.
25.	Які із вказаних тверджень є істинними стосовно оператора break? 1) тіло циклу виконується ще один раз; 2) відбувається негайний перехід до наступної ітерації; 3) цикл переривається; 4) відбувається зупинка циклу на вказану кількість мілісекунд; 5) дія залежить від умови, що вказано при описі циклу; 6) дія залежить від умови, що вказано безпосередньо після оператора break; 7) немає правильної відповіді.
26.	Які із вказаних тверджень є істинними стосовно оператора continue? 1) тіло циклу виконується ще один раз; 2) відбувається негайний перехід до наступної ітерації; 3) цикл переривається;

- | |
|---|
| 4) відбувається продовження циклу після зупинки на вказану кількість мілісекунд;
5) дія залежить від умови, що вказано при описі циклу;
6) дія залежить від умови, що вказано безпосередньо після оператора <code>continue</code> ;
7) немає правильної відповіді. |
|---|

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. C# Tutorial. 2023. URL: <https://www.javatpoint.com/c-sharp-tutorial>
2. Learn C# Programming. 2023. URL: <https://www.tutorialsteacher.com/csharp>
3. C#.NET Tutorials For Beginners and Professionals. 2023. URL: <https://dotnettutorials.net/course/csharp-dot-net-tutorials/>

ЗМІСТ

РЕАЛІЗАЦІЯ ЦИКЛІЧНИХ АЛГОРИТМІВ	3
Оператори циклу	3
Оператори break та continue.....	6
НАБІР ТЕСТІВ ДЛЯ САМОКОНТРОЛЮ	9
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	13

Автори: к. ф.-м. н., доц. Брила А.Ю.,
ст.викл., Ломага М.М.
к. ф.-м. н. Антосяк П.П.

Рецензенти: к.ф.-м.н., доц. Погоріляк О.О.,
к.т.н., доц. Андрашко Ю.В.

РЕАЛІЗАЦІЯ ЦИКЛІЧНИХ АЛГОРИТМІВ У С#

Методичні матеріали з організації самостійної роботи студентів з дисципліни
“Програмування”