

РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМ

Повна форма

Умовний оператор

Скорочена форма

Програмна структура	Аналог на мові блок-схем	Приклад
<pre>if (<умова>) <оператор1>; else <оператор2>;</pre>		<pre>if(x>y) max=x; else max=y;</pre>

Програмна структура	Аналог на мові блок-схем	Приклад
<pre>if (<умова>) <оператор1>;</pre>		<pre>if(x!=0) z=1/x;</pre>

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            // введення вихідних даних
            Console.WriteLine("x=");
            double x = double.Parse(Console.ReadLine());
            // реалізація алгоритму розв'язання задачі
            if (x >= 0 && x <= 1)
                Console.WriteLine("належить");
            else
                Console.WriteLine("не належить");
            //
            Console.ReadKey();
        }
    }
}
```

Визначити чи належить значення дійсної змінної x проміжку $[0,1]$ та вивести відповідне повідомлення

$x=0,3$

належить

```
1  int num1 = 8;
2  int num2 = 6;
3  if(num1 > num2)
4  {
5      Console.WriteLine($"Число {num1} больше числа {num2}");
6  }
7  else if (num1 < num2)
8  {
9      Console.WriteLine($"Число {num1} меньше числа {num2}");
10 }
11 else
12 {
13     Console.WriteLine("Число num1 равно числу num2");
14 }
```

РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМ

Скласти програму для обчислення значення функції

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            // введення вихідних даних
            Console.Write("x=");
            double x = double.Parse(Console.ReadLine());
            // реалізація алгоритму розв'язання задачі
            double f;
            if (x <= 0)
                f = 0;
            else
                if (x <= 1)
                    f = x * x - x;
                else
                    f = x * x - Math.Sin(Math.PI * x * x);
            // вивід результату
            Console.WriteLine("f={0}", f);
            //
            Console.ReadKey();
        }
    }
}
```

$$f(x) = \begin{cases} 0, & \text{якщо } x \leq 0, \\ x^2 - x, & \text{якщо } 0 < x \leq 1, \\ x^2 - \sin \pi x^2, & \text{якщо } x > 1. \end{cases}$$

$x=0,5$

$f=-0,25$

РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМ

Складений оператор

Повна форма

Програмна структура	Аналог на мові блок-схем	Приклад
<pre>if (<умова>) { <оператор1.1>; <оператор1.N>; } else { <оператор2.1>; <оператор2.M>; }</pre>		<pre>if(x>y) { max=x; min=y; } else { max=y; min=x; }</pre>

Скорочена форма

Програмна структура	Аналог на мові блок-схем	Приклад
<pre>if (<умова>) { <оператор1>; <операторN>; }</pre>		<pre>if(x>0) { z=1/x; l=y/x; }</pre>

Знайти максимальне та мінімальне із двох дійсних чисел

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
namespace ConsoleApplication1  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            double b,d;  
            Console.Write("b=");  
            b = double.Parse(Console.ReadLine());  
            Console.Write("d=");  
            d = double.Parse(Console.ReadLine());  
            double max, min;  
            if (b > d)  
            {  
                max = b; min = d;  
            }  
            else  
            {  
                max = d; min = b;  
            }  
            Console.WriteLine("max={0} min={1}", max, min);  
            Console.ReadLine();  
        }  
    }  
}
```

b=3
d=8

max=8 min=3

РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМ

Умовний оператор (?:)

Загальний вигляд умовного оператора (?:)	Аналог з використанням умовного оператора if	З використанням умовного оператора (?:)	
<code><змінна>=<умова>?<значення1>:<значення2>;</code>	<code>if (<умова>) <змінна>=<значення1>; else <змінна>=<значення2>;</code>	<code>max = (x>y) ? x : y;</code>	<code>if (x>y) max=x; else max=y;</code>

```
1 int x=3;  
2 int y=2;  
3 Console.WriteLine("Нажмите + или -");  
4 string selection = Console.ReadLine();  
5  
6 int z = selection=="+"? (x+y) : (x-y);  
7 Console.WriteLine(z);
```

*У якості селектора вибору може бути вираз типу **int**, типу **char**, типу **enum** або типу **string***

Загальне правило запису	Приклад
<p>Оператор вибору switch</p> <pre>switch (<селектор вибору>) { case <константа1> : <оператор1>; break; case <константа2> : <оператор2>; break; case <константаN> : <операторN>; break; default : <оператор N+1>; break; }</pre>	<p>Вводиться оцінка – цифра, вивести оцінку прописом (селектор вибору цілого типу).</p> <pre>using System; using System.Collections.Generic; using System.Linq; using System.Text; namespace ConsoleApplication1 { class Program { static void Main(string[] args) { int mark; Console.Write("Mark = "); mark = int.Parse(Console.ReadLine()); switch (mark) { case 2: Console.WriteLine("Незадовільно"); break; case 3: Console.WriteLine("Задовільно."); break; case 4: Console.WriteLine("Добре"); break; case 5: Console.WriteLine("Відмінно"); break; default: Console.WriteLine("Неправильна оцінка."); break; } Console.ReadKey(); } } }</pre>

РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМ

Оператор вибору switch

З клавіатури вводиться оцінка у національній шкалі, необхідно вивести повідомлення про те, чи зараховано студенту залік

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int mark;
            Console.Write("Mark = ");
            mark = int.Parse(Console.ReadLine());
            switch (mark)
            {
                case 1:
                case 2: Console.WriteLine("Незараховано.");
                    break;
                case 3:
                case 4:
                case 5: Console.WriteLine("Зараховано.");
                    break;
                default: Console.WriteLine("Неправильна оцінка.");
                    break;
            }
            Console.ReadKey();
        }
    }
}
```

РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМ

Оператор вибору switch

З клавіатури вводиться колір помідора, на екран необхідно вивести у якому він стані (росте, дозріває чи можна зірвати). Селектор і константи вибору є величинами типу string

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            string color;
            Console.Write("Color = ");
            color = Console.ReadLine();
            switch (color)
            {
                case "Green" : Console.WriteLine("Ще росте");
                    break;
                case "Yellow": Console.WriteLine("Дозріває.");
                    break;
                case "Red": Console.WriteLine("Можна зірвати");
                    break;
                default: Console.WriteLine("Помідор-мутант.");
                    break;
            }
            Console.ReadKey();
        }
    }
}
```

РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМ

Оператор вибору switch

З клавіатури вводиться буква у нижньому регістрі, з'ясувати, чи є буква голосною. Селектор і константи вибору є величинами типу char

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            char c;
            Console.Write("Input letter= ");
            c = (char)Console.Read();
            switch (c)
            {
                case 'a' :
                case 'o' :
                case 'y' :
                case 'и' :
                case 'i' :
                case 'e' : Console.WriteLine("Голосна");
                           break;
                default: Console.WriteLine("Приголосна");
                           break;
            }
            Console.ReadKey();
        }
    }
}
```

РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМ

Оператор вибору switch

У кінці кожного блоку **case** повинен бути один з операторів переходу: **break**, **goto case**, **return** або **throw**. Як правило, використовується оператор **break**. При його застосуванні інші блоки **case** виконуватись не будуть.

Однак якщо ми хочемо, щоб, навпаки, після виконання поточного блоку **case** виконувався інший блок **case**, то можемо використовувати замість **break** оператор **goto case**:

```
1  int number = 1;
2  switch (number)
3  {
4      case 1:
5          Console.WriteLine("case 1");
6          goto case 5; // переход к case 5
7      case 3:
8          Console.WriteLine("case 3");
9          break;
10     case 5:
11         Console.WriteLine("case 5");
12         break;
13     default:
14         Console.WriteLine("default");
15         break;
16 }
```


РЕАЛІЗАЦІЯ ЦИКЛІЧНИХ АЛГОРИТМІВ

Оператор циклу while

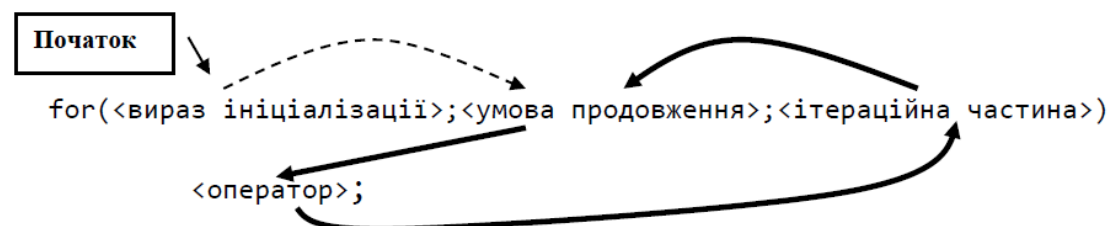
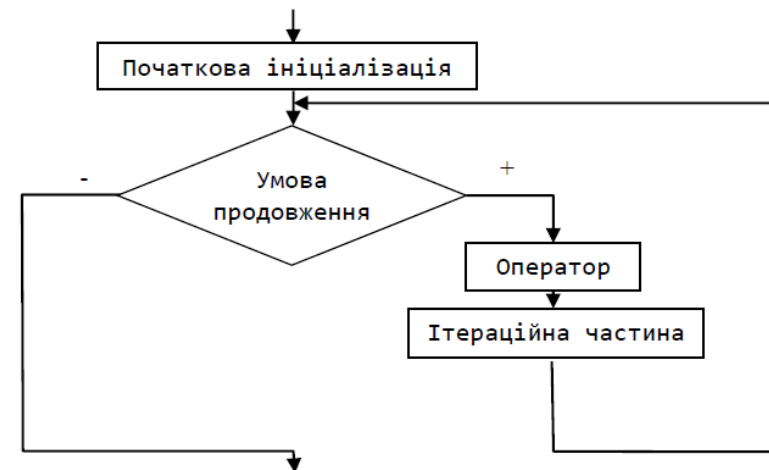
Програмна структура	Аналог на мові блок-схем	Приклад. Знайти суму перших n натуральних чисел.
<pre>while (<умова>) <оператор>;</pre>		<pre>int sum=0; int i=1; while(i<=n) sum=sum+i++;</pre>

Оператор циклу do-while

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>

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 for



РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМ

Оператор циклу for

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();
        }
    }
}
```

Обчислити значення виразу

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

```
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 нескінченний цикл може бути задано так:

```
for ( ; ; )
    <оператор>;
```

РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМ

Оператори break та continue

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();
        }
    }
}
```

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();
        }
    }
}
```

Оператор **continue** може бути використаний у будь-якому із циклів у випадку, коли немає потреби виконувати до кінця усі оператори тіла циклу поточної ітерації, а необхідно одразу перейти до наступної ітерації.

РЕАЛІЗАЦІЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМ

Оператори break та continue

```
1 for (int i = 0; i < 9; i++)
2 {
3     if (i == 5)
4         break;
5     Console.WriteLine(i);
6 }
```

```
0
1
2
3
4
```

```
1 for (int i = 0; i < 9; i++)
2 {
3     if (i == 5)
4         continue;
5     Console.WriteLine(i);
6 }
```

```
0
1
2
3
4
6
7
8
```

Цикл **foreach...in** – спеціальний цикл для роботи з набором елементів – колекцією (прикладом одного з типів колекцій є масив). Має такий синтаксис:

foreach(<тип> <ім'я_змінної> in <вираз_колекція>) оператор;

```
int[] myArray = { 1, 3, 5, 8, 2 };
foreach (int i in myArray) {
    Console.WriteLine("i = " + i);
}
```