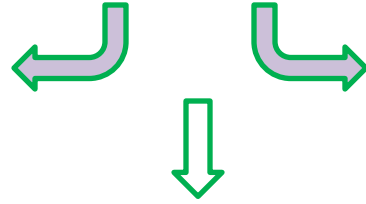


# Операторы управления

## Линейный

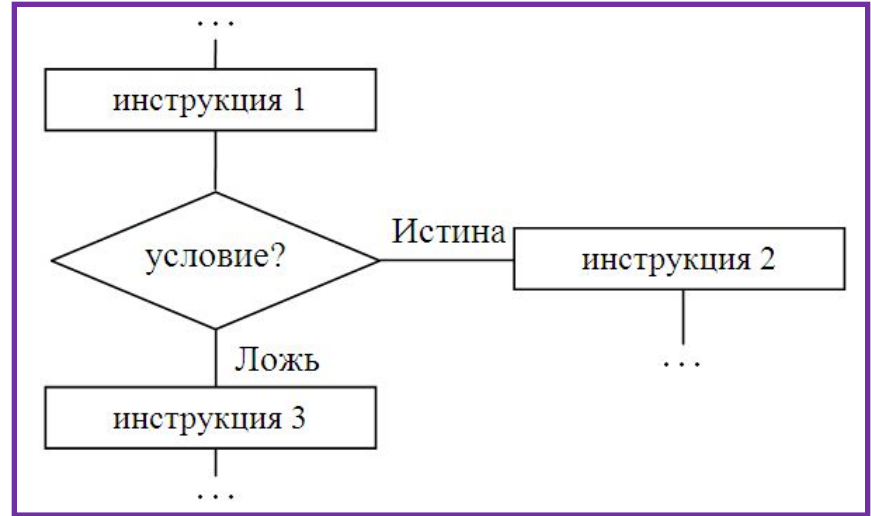


## Типы вычислительных процессов

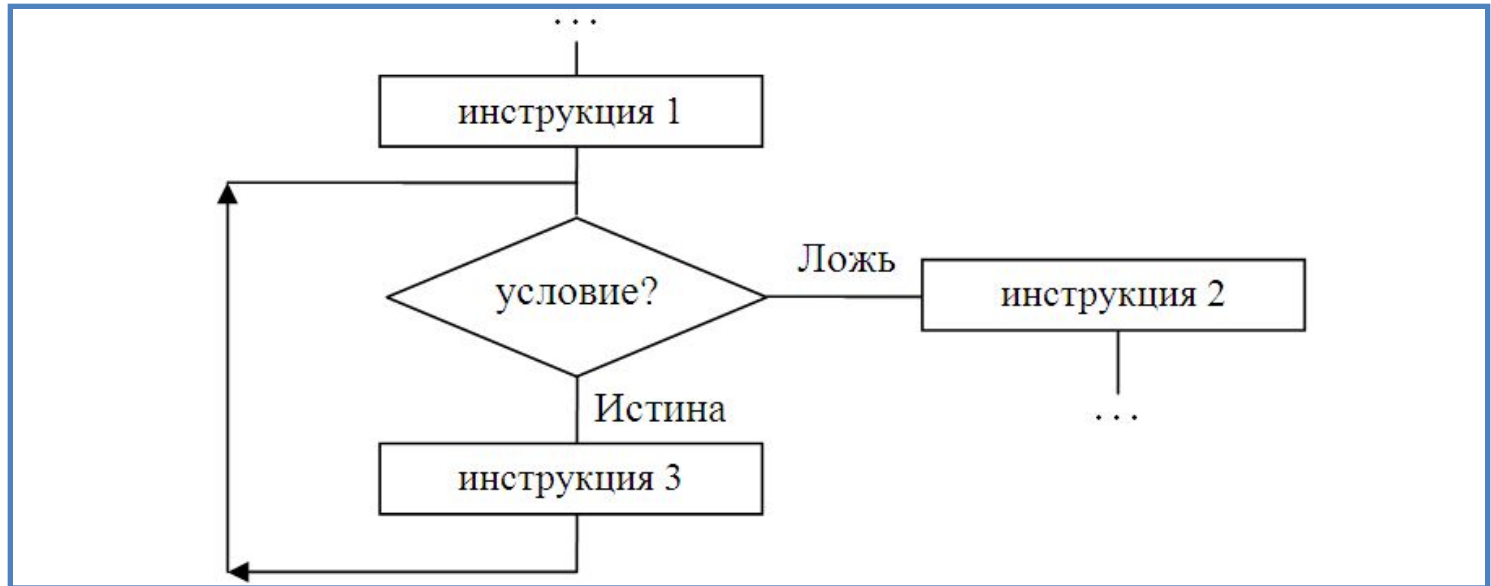


## Циклический

## Ветвления



*Циклы в программах составляют, в среднем, 80–85% общего объёма кода*



# Операторы выбора

## Оператор if

```
if (выражение) operator1;  
operator2;
```

значение в скобках булевого типа



```
using System;  
{  
    static void Main()  
    {  
        int i, j, k;  
        string s;  
        Console.WriteLine("Задайте первое число!");  
        s = Console.ReadLine();  
        i = Convert.ToInt32(s);  
        Console.WriteLine("Задайте второе число!");  
        s = Console.ReadLine();  
        j = Convert.ToInt32(s);  
        if (i < j) k = j;  
        if (i > j) k = i;  
        Console.WriteLine("Максимум из заданных чисел = {0}",k);  
    }  
}
```

## Оператор *if - else*

```
if (выражение) operator1;  
  else operator2;  
operator3;
```

```
using System;  
{  
  static void Main()  
  {
```

```
    int i, j, k;
```

```
    string s;
```

```
    Console.WriteLine("Задайте первое число!");
```

```
    s = Console.ReadLine();
```

```
    i = Convert.ToInt32 ( s );
```

```
    Console.WriteLine ("Задайте второе число!" );
```

```
    s = Console.ReadLine();
```

```
    j = Convert.ToInt32 ( s );
```

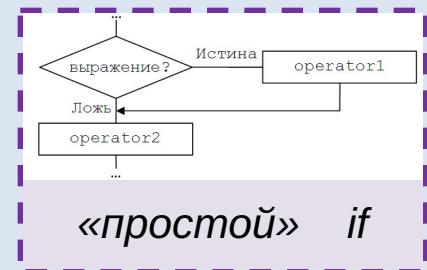
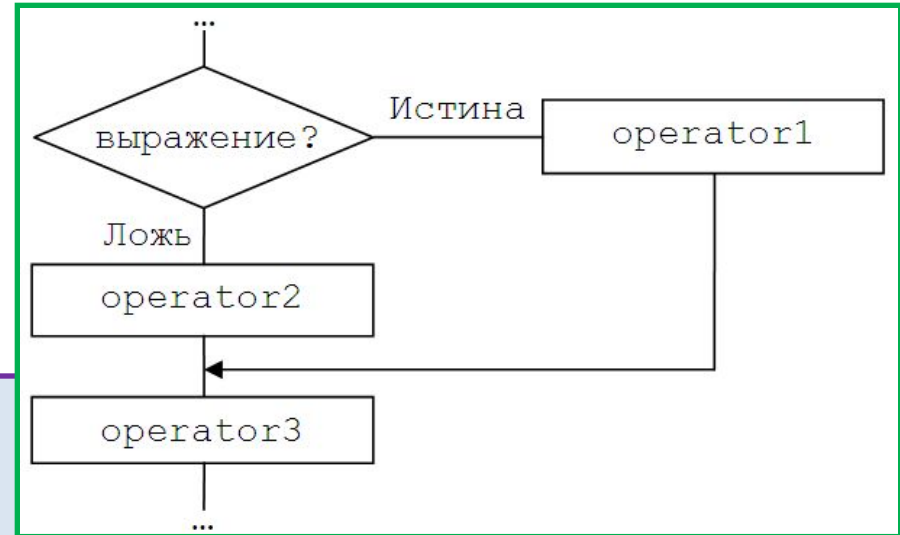
```
    if ( i > j ) k = i;
```

```
        else k = j;
```

```
    Console.WriteLine ("Максимум из заданных чисел = {0}", k );
```

```
  }
```

```
}
```



## Оператор *if-else-if*

```
if (выражение1) оператор1;  
    else if (выражение2) оператор2;  
    else оператор3;  
оператор4;
```

```
using System;  
class Primer  
{
```

```
    static void Main()  
    {
```

```
        int x, f;
```

```
        string s;
```

```
        Console.WriteLine("\t\tРаботает сигнальная функция");
```

```
        Console.WriteLine("Задайте значение x!");
```

```
        s = Console.ReadLine();
```

```
        x = Convert.ToInt32(s);
```

```
        if (x < 0) f = -1;
```

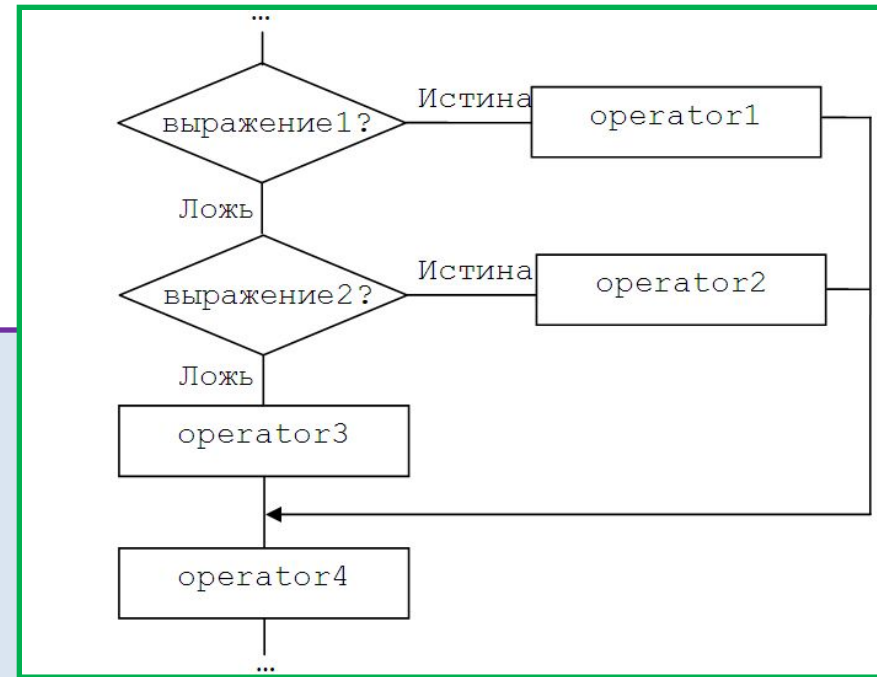
```
            else if (x > 0) f = 1;
```

```
                else f = 0;
```

```
        Console.WriteLine("Значение сигнальной функции = {0}", f);
```

```
    }
```

```
}
```



$$f(x) = \begin{cases} 1, & \text{для всех } x > 0 \\ 0, & \text{для } x = 0 \\ -1, & \text{для всех } x < 0 \end{cases}$$

```
Работает сигнальная функция  
Задайте значение x!  
100  
Значение сигнальной функции = 1
```

## Блочный оператор

1. **Блочный оператор (или блок)** – это инструкции (операторы), размещённые внутри парных фигурных скобок.
2. Самостоятельное значение имеют блоки в операторах выбора и циклов.
3. В **операторах выбора** (а также – циклов) на месте operator может находиться блок операторов, инструкции в котором будут выполнены в соответствии с рассматриваемой логикой

```
using System;
class Primer
{ static void Main()
  { int i, j, max, min;
    string s;
    Console.WriteLine("Задайте первое число!");
    s = Console.ReadLine();
    i = Convert.ToInt32(s);
    Console.WriteLine("Задайте второе число!");
    s = Console.ReadLine();
    j = Convert.ToInt32(s);
    if (i > j) { max = i; min = j; }
      else { max = j; min = i; }
    Console.WriteLine("Максимальное = {0}, минимальное = {1} ",max, min);
  }
}
```

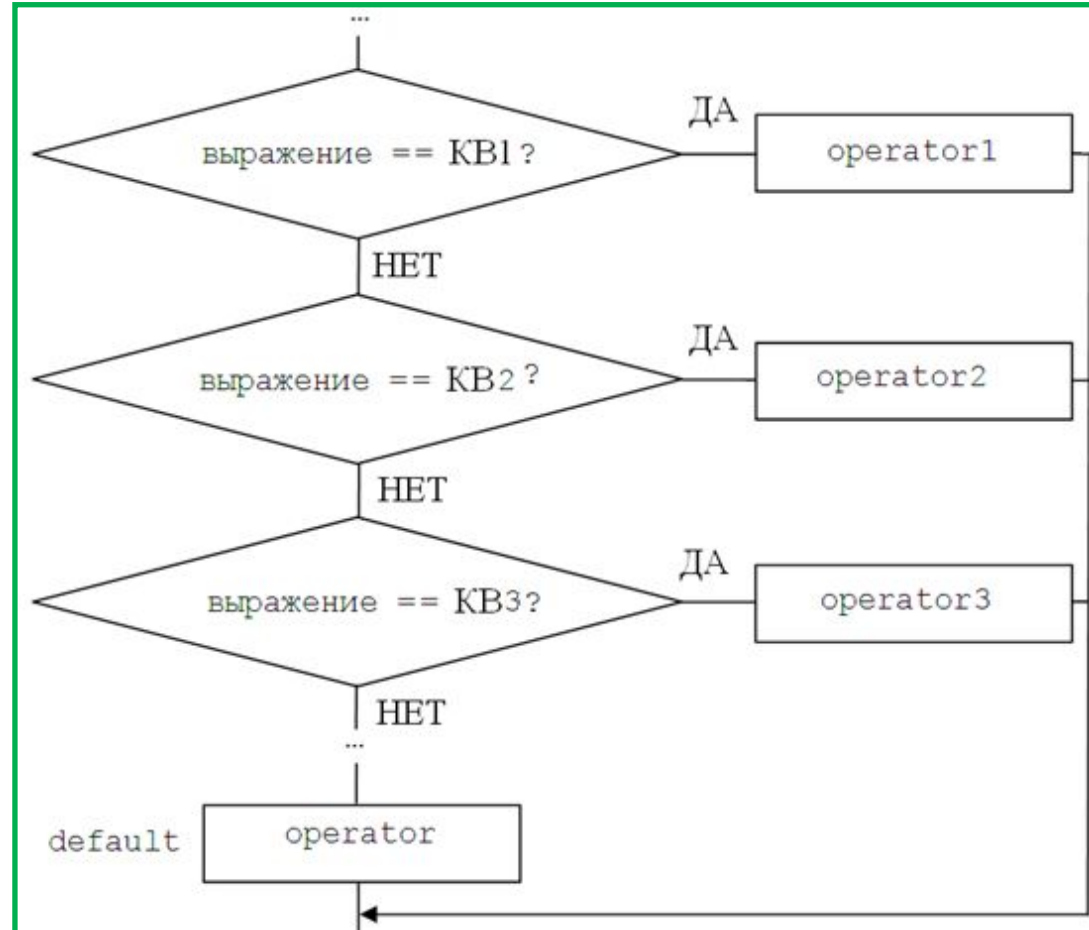
## Оператор множественного выбора switch

- выполняет те же действия, что и многоступенчатый if else, но более наглядным образом

```
switch (выражение)
{
case KB1 : operator1; break;
case KB2 : operator2; break;
case KB3 : operator3; break;
...
default: operator; break;
}
```

**KB** – константное выражение. Обычно вместо него используется целочисленная или строковая **константа**.

На месте любого оператора может быть последовательность операторов (раздел). Заканчиваться раздел должен оператором перехода. Обычно используют **break**, действием которого является передача управления за границу блока **switch**



Метка default помечает раздел, который выполняется, если не было вхождения в какую-либо метку с **KB**

## Примеры на switch

```
using System;
class Primer2
{
    static void Main()
    {
        int x= 0xd;
        switch(x)
        {
            default: x += 1; goto case 3;
            case 1: x += 2; break;
            case 2: x += 3; goto case 1;
            case 3:
            case 4: x += 4; goto case 2;
            case 5: x += 5; break;
            case 6: x += 6; break;
        }
        Console.WriteLine("x= {0} ", x);
    }
}
```

x = 23

метку **default**: можно размещать в любом месте **switch**, но только не после пустой метки !  
( в этом примере нельзя после case 3: )

```
using System;
class Primer1
```

x = 8

```
{
    static void Main()
    {
        int x=3;
        switch(x)
        {
            case 2: x+=2; break;
            case 1: x+=1; break;
            case 3:
            case 4:
            case 5: x+=5; break;
            default: x-=10; break;
        }
        Console.WriteLine("x = {0}", x);
    }
}
```

Оператор **goto** осуществляет переход на **метку**, имя которой указывается в качестве его параметра.

**Метка** может располагаться или в **том же**, или в более **внешнем блоке**.

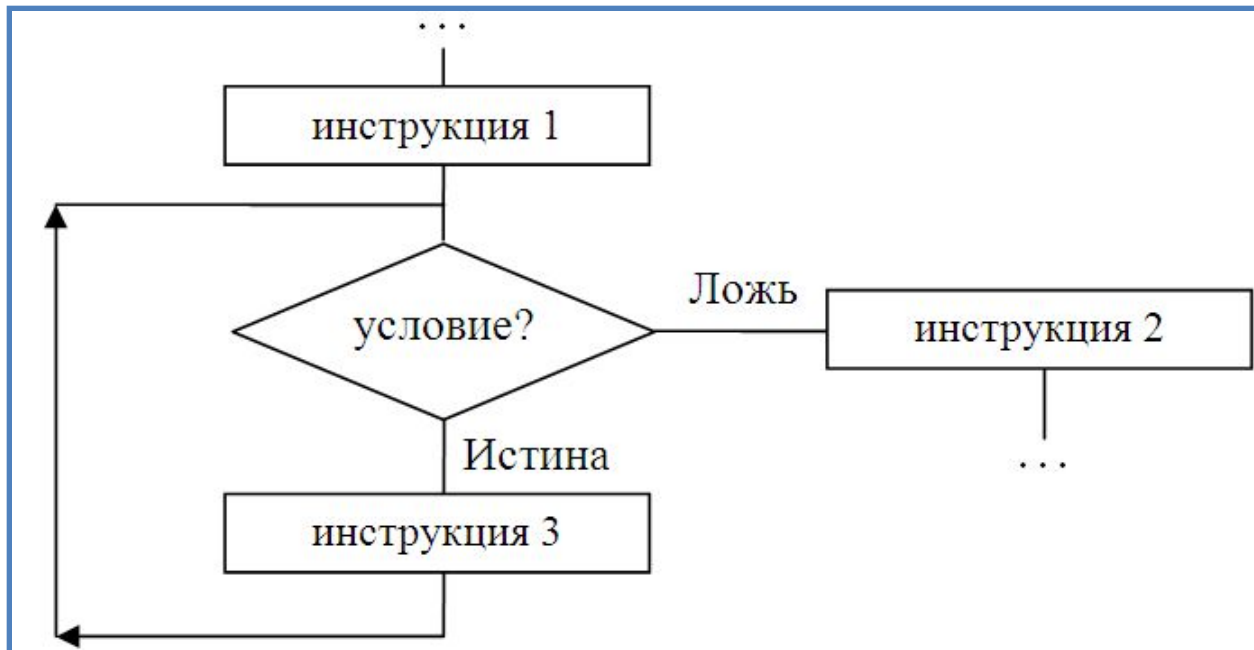
# Операторы циклов

с предусловием

**Типы циклов**

с постусловием

*инструкция 3* – это т.н. «тело цикла»





## Оператор цикла while

```
while (выражение) оператор1;  
оператор2;
```

на месте «оператор1» может быть:

- простой,
- пустой,
- блочный оператор.



```
using System;  
class Primer  
{ static void Main()  
    { uint i, j = 1;  
      uint f = 1;  
      string s;  
      Console.WriteLine("Задайте натуральное число!");  
      s = Console.ReadLine();  
      i = Convert.ToUInt32(s);  
      while (j <= i)  
          { f *= j;  
            j++;  
          }  
      Console.WriteLine("Факториал от {0} = {1}", i, f);  
    }  
}
```

```
Задайте натуральное число!  
18  
Факториал от 18 = 3396534272  
Задайте натуральное число!  
19  
Факториал от 19 = 109641728
```

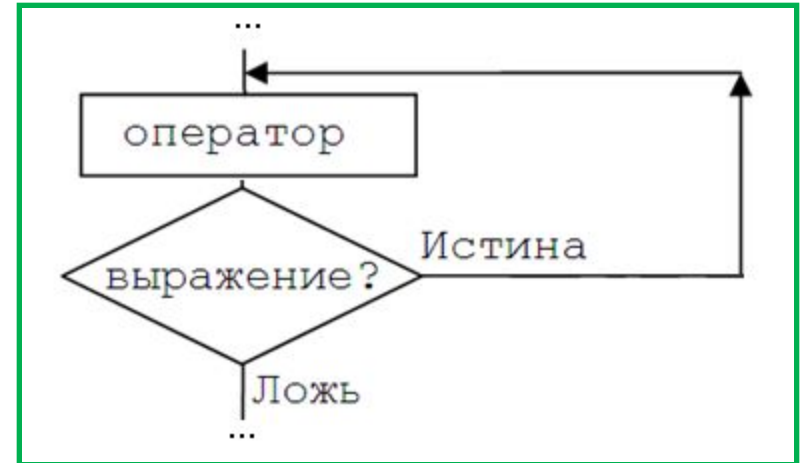
Справочно:  $\max_{uint} = 4294967295$

## Оператор цикла do while

*do оператор;*  
*while выражение;*

```
using System;  
class Primer  
{  
    static void Main()  
    {  
        int num= 12345, next;  
        do  
        {  
            next = num % 10;  
            Console.Write( next );  
            num = num / 10;  
        }  
        while ( num > 0 );  
    }  
}
```

54321



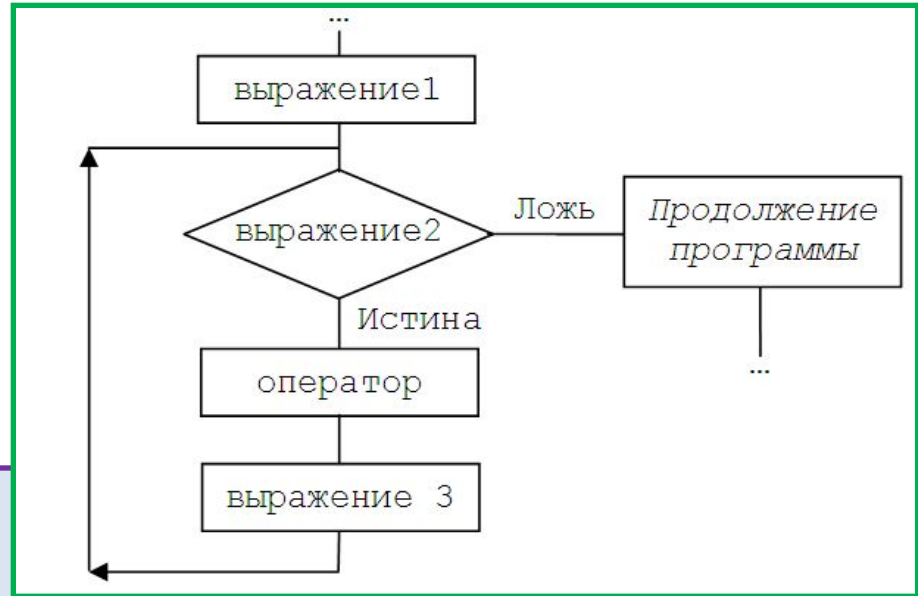
# Оператор цикла for

`for (выражение1; выражение2; выражение3) оператор ;`

*Является универсальным и представляет собой своего рода шаблон для типичного цикла*

*for (инициализация; условие; итерация)  
оператор ;*

```
using System;
class Primer
{ static void Main()
  { uint i, j;
    uint f;
    string s;
    Console.WriteLine("Задайте натуральное число!");
    s = Console.ReadLine();
    i = Convert.ToUInt32( s );
    for (f = j = 1; j <= i; j++) f *= j;
    Console.WriteLine("{0}! = {1}", i, f);
  }
}
```



**формально:  
пустой цикл**

`for (f = j = 1; j <= i; f*=j++);`

**бесконечный и  
пустой цикл:  
for (;;) ;**

## Цикл for (пример)

```
using System;
class Primer
{
    static void Main()
    {
        int i, sum = 0 ;
        for (i = 1; i < 10; i+=2, sum++) ;
        Console.WriteLine("i={0}, sum = {1}", i, sum);
    }
}
```

*i=11, sum = 5*

## Цикл for (пример)

```
using System;
class Primer
{
    static void Main()
    {
        int i, j, k = 0 ;
        for ( i = 0, j = 10; i <= j; i++, j--) k++;
        Console.WriteLine("i={0}, j = {1}, k = {2}", i, j, k );
    }
}
```

$i = 6, j = 4, k = 6$

## Операторы goto, break и continue в циклах (пример1)

```
using System;
class Primer
{ static void Main( )
  { short i, j;
    short f ;
    string s;
    Console.WriteLine("Задайте натуральное число!");
    s = Console.ReadLine( );
    i = Convert.ToInt16 ( s );
    for (f = j = 1; j <= i; j++)
    {
      f *= j;
      if ( j == 7 ) break;
    }
    Console.WriteLine("{0}! = {1}", i, f );
  }
}
```

```
Задайте натуральное число?
6
6! = 720
Задайте натуральное число?
7
7! = 5040
Задайте натуральное число?
8
8! = 5040
```

$$8! = 40320$$

$$\max_{\text{short}} = 2^{15} - 1 = 32767$$



## Операторы goto, break и continue в циклах (продолжение примера1)

```
using System;
class Primer
{
    static void Main ( )
    {
        short i, j;
        short f;
        string s;
        Console.WriteLine( "Задайте натуральное число!" );
        s = Console.ReadLine ( );
        i = Convert.ToInt16 ( s );
        for (f = j = 1; j <= i; j++)
        {
            f *= j;
            if (j == 7) goto m1;
        }
        Console.WriteLine("{0}! = {1}", i, f);
        return;
m1: Console.WriteLine("Наибольшее {0}! = {1}", j, f);
    }
}
```

Задайте натуральное число!  
6  
6! = 720

Задайте натуральное число!  
7  
Наибольшее 7! = 5040

Задайте натуральное число!  
10  
Наибольшее 7! = 5040

## Операторы goto, break и continue в циклах (ещё один вариант примера1)

```
using System;
class Primer
{
    static void Main()
    {
        short i, j;
        short f;
        string s;
        Console.WriteLine ( "Задайте натуральное число!" );
        s = Console.ReadLine ( );
        i = Convert.ToInt16 ( s );
        for (f = j = 1; j <= i; j++)
        {
            if ( j > 7 ) continue;
            f *= j;
        }
        Console.WriteLine("{0}! = {1}", (i<=7) ? i : (short)7, f);
    }
}
```

```
Задайте натуральное число!
10
7! = 5040
```

## Операторы `goto`, `break` и `continue` в циклах (пример2)

```
using System;
class Primer
{ static void Main()
  { short i, j;
    short f = 0;
    string s;
    Console.WriteLine("Задайте натуральное число!");
    s = Console.ReadLine();
    i = Convert.ToInt16(s);
    for ( j = 1; j <= i; j++)
    {
      if ( j%2 == 0) continue;
      f += j;
    }
    Console.WriteLine("Сумма нечётных интервала 1 - {0}= {1}", j, f);
  }
}
```

Задайте натуральное число?

10

Сумма нечётных интервала 1 - 11 = 25

## Вложенные циклы

```
using System;
class Primer
{
    static void Main()
    {
        for (short j = 1; j <= 5; Console.WriteLine( ), j++)
            for (short i = 1; i < 5; i++)
                Console.Write(" \t{0}", i * j);
    }
}
```

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16
5	10	15	20

```
using System;
class Primer
{
    static void Main()
    {
        for (short j = 1; j <= 5; Console.WriteLine(), j++)
            for (short i = 1; i <= j; i++)
                Console.Write("{0,5}", i * j);
    }
}
```

1				
2	4			
3	6	9		
4	8	12	16	
5	10	15	20	25