

## Практическая работа №4

### Операции и выражения. Правила формирования и вычисления выражений.

Цель работы: Ознакомиться с правилами записи операций и выражений в языке программирования C#, получить основные навыки работы с ней, освоить приёмы создания, компиляции и исполнения программы.

Норма времени: 2 часа.

Оборудование: персональный компьютер, система программирования C#.

### Ход работы

#### Операции

##### 1. Инкремент (++) и декремент(--).

Эти операции имеют две формы записи - *префиксную*, когда операция записывается перед операндом, и *постфиксную* - операция записывается после операнда. *Префиксная операция* инкремента (декремента) увеличивает (уменьшает) свой операнд и возвращает измененное значение как результат. Постфиксные версии инкремента и декремента возвращают первоначальное значение операнда, а затем изменяют его.

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

namespace ConsoleApplication4
{
    class Program
    {
        static void Main()
        {
            int i = 3, j = 4;
            Console.WriteLine("{0} {1}", i, j);
            Console.WriteLine("{0} {1}", ++i, --j);
            Console.WriteLine("{0} {1}", i++, j--);
            Console.WriteLine("{0} {1}", i, j);
            Console.ReadKey();
        }
    }
}
```

**Замечание.** Префиксная версия требует существенно меньше действий: она изменяет значение переменной и запоминает результат в ту же переменную. Постфиксная операция должна отдельно сохранить исходное значение, чтобы затем вернуть его как результат. Для сложных типов подобные дополнительные действия могут оказаться трудоемки. Поэтому постфиксную форму имеет смысл использовать только при необходимости.

##### 2. Отрицание:

- Арифметическое отрицание (-) - меняет знак операнда на противоположный.
- Логическое отрицание (!) - определяет операцию инверсия для логического типа.

```
using System;
using System.Collections.Generic;
```

```

using System.Linq;
using System.Text;

namespace ConsoleApplication5
{
    class Program
    {
        static void Main()
        {
            int i = 3, j = -4;
            bool a = true, b = false;
            Console.WriteLine("{0} {1}", -i, -j);
            Console.WriteLine("{0} {1}", !a, !b);
            Console.ReadKey();
        }
    }
}

```

**3. Умножение (\*), деление (/) и деление с остатком (%).** Операции умножения и деления применимы для целочисленных и вещественных типов данных. Для других типов эти операции применимы, если для них возможно *неявное преобразование* к целым или вещественным типам. При этом тип результата равен "наибольшему" из типов операндов, но не менее int. Если оба операнда при делении целочисленные, то и результат тоже целочисленный.

```

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

namespace ConsoleApplication7
{
    class Program
    {
        static void Main()
        {
            int i = 100, j = 15;
            double a = 14.2, b = 3.5;
            Console.WriteLine("{0} {1} {2}", i * j, i / j, i % j);
            Console.WriteLine("{0} {1} {2}", a * b, a / b, a % b);
            Console.ReadKey();
        }
    }
}

```

**4. Сложение (+) и вычитание (-).** Операции сложения и вычитания применимы для целочисленных и вещественных типов данных. Для других типов эти операции применимы, если для них возможно *неявное преобразование* к целым или вещественным типам.

**5. Операции отношения (<, <=, >, >=, ==, !=).** Операции отношения сравнивают значения левого и правого операндов. Результат операции логического типа: true - если значения совпадают, false - в противном случае.

```

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

namespace ConsoleApplication8

```

```

{
    class Program
    {
        static void Main()
        {
            int i = 15, j = 15;
            Console.WriteLine(i < j); //меньше
            Console.WriteLine(i <= j); //меньше или равно
            Console.WriteLine(i > j); //больше
            Console.WriteLine(i >= j); //больше или равно
            Console.WriteLine(i == j); //равно
            Console.WriteLine(i != j); //не равно
            Console.ReadKey();
        }
    }
}

```

Логические операции применяются к операндам логического типа.

Результат логической операции **И** имеет значение истина тогда и только тогда, когда оба операнда принимают значение истина.

Результат логической операции **ИЛИ** имеет значение истина тогда и только тогда, когда хотя бы один из операндов принимает значение истина.

```

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

```

```

namespace ConsoleApplication9

```

```

{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("x    y    x и y    x или y");
            Console.WriteLine("{0} {1} {2} {3}", false, false, false && false, false || false);
            Console.WriteLine("{0} {1} {2} {3}", false, true, false && true, false || true);
            Console.WriteLine("{0} {1} {2} {3}", true, false, true && false, true || false);
            Console.WriteLine("{0} {1} {2} {3}", true, true, true && true, true || true);
            Console.ReadKey();
        }
    }
}

```

## 6. Условная операция.

Формат: (<операнд1>)? <операнд2> : <операнд3> ;

Операнд1 - это логическое выражение, которое оценивается с точки зрения его эквивалентности константам true и false. Если результат вычисления операнда1 равен true, то результатом условной операции будет значение операнда2, иначе - операнда3. Фактически условная операция является сокращенной формой условного оператора if, который будет рассмотрен позже.

```

static void Main()
{
    int x=5; int y=10;
    int max = (x > y) ? x : y;
    Console.WriteLine(max);
}

```

**7. Операции присваивания:** =, +=, -= и т.д.

Формат операции *простого присваивания* ( = ):

операнд\_2 = операнд\_1;

В результате выполнения этой операции вычисляется значение операнда\_1, и результат записывается в операнд\_2. Возможно связать воедино сразу несколько операторов присваивания, записывая такие цепочки: a=b=c=100. Выражение такого вида выполняется справа налево: результатом выполнения c=100 является число 100, которое затем присваивается переменной b, результатом чего опять является 100, которое присваивается переменной a.

Кроме простой операции присваивания существуют *сложные операции присваивания*, например, умножение с присваиванием ( \*= ), деление с присваиванием ( /= ), остаток от деления с присваиванием ( %= ), сложение с присваиванием ( += ), вычитание с присваиванием ( -= ) и т.д.

В сложных операциях присваивания, например, при *сложении с присваиванием*, к операнду\_2 прибавляется операнд\_1, и результат записывается в операнд\_2. То есть, выражение c += a является более компактной записью выражения c = c + a. Кроме того, сложные операции присваивания позволяют сгенерировать более эффективный код, за счет того, что в простой операции присваивания для хранения значения правого операнда создается временная переменная, а в сложных операциях присваивания значение правого операнда сразу записывается в левый операнд.

Рассмотренные *операции* приведены с учетом убывания приоритета. Если в одном выражении соседствуют *операции* одного приоритета, то *операции* присваивания и условная *операции* выполняются справа налево, а остальные наоборот. Если необходимо изменить порядок выполнения операций, то в выражении необходимо поставить круглые скобки.

### **Выражения и преобразование типов**

*Выражение* - это синтаксическая *единица* языка, определяющая способ вычисления некоторого значения. Выражения состоят из операндов, операций и скобок. Каждый *операнд* является в свою *очередь* выражением или одним из его частных случаев - константой, переменной или функций.

(a + 0.12)/6

x && y || !z

(t \* Math.Sin(x)-1.05e4)/((2 \* k + 2) \* (2 \* k + 3))

*Операции* выполняются в соответствии с приоритетами. Для изменения порядка выполнения операций используются круглые скобки. Если в одном выражении записано несколько операций одинакового приоритета, то унарные *операции*, условная операция и *операции* присваивания выполняются *справа налево*, остальные - *слева направо*. Например,

a = b = c означает a=(b=c),

a+b+c означает (a + b) + c.

В *выражение* могут входить операнды различных типов. Если операнды имеют одинаковый тип, то результат *операции* будет иметь тот же тип. Если

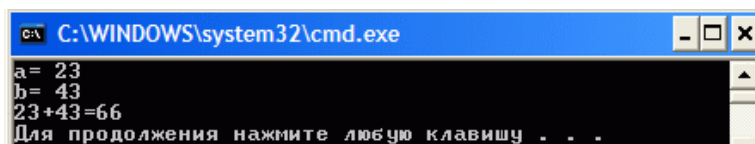
операнды разного типа, то перед вычислениями выполняются преобразования более коротких типов в более длинные для сохранения значимости и точности.

## Индивидуальные задания:

### Вариант 1.

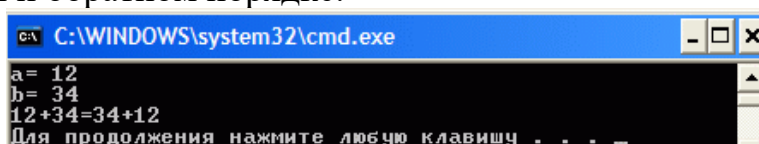
**Задание.** Написать программу, которая реализует диалог с пользователем:

1. запрашивает с клавиатуры два целых числа, и выводит на экран сумму данных чисел:



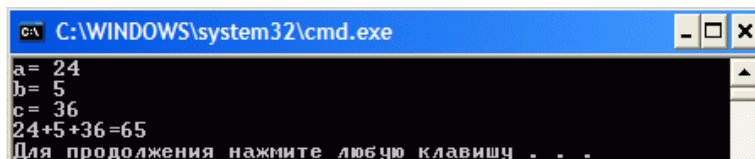
```
C:\WINDOWS\system32\cmd.exe
a= 23
b= 43
23+43=66
Для продолжения нажмите любую клавишу . . .
```

2. запрашивает с клавиатуры три целых числа, и выводит на экран сумму данных чисел в прямом и обратном порядке:



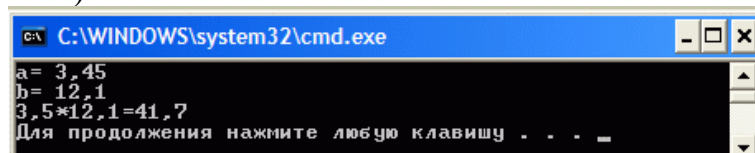
```
C:\WINDOWS\system32\cmd.exe
a= 12
b= 34
12+34=34+12
Для продолжения нажмите любую клавишу . . .
```

3. запрашивает с клавиатуры три целых числа, и выводит на экран сумму данных чисел:



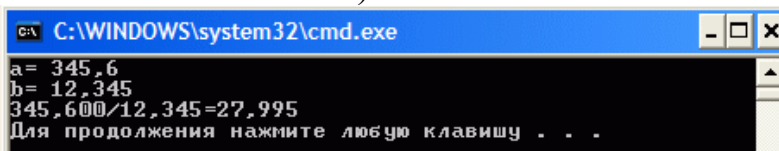
```
C:\WINDOWS\system32\cmd.exe
a= 24
b= 5
c= 36
24+5+36=65
Для продолжения нажмите любую клавишу . . .
```

4. запрашивает с клавиатуры два вещественных числа, и выводит на экран произведение данных чисел (вещественные числа выводятся с точностью до 1 знака после запятой):



```
C:\WINDOWS\system32\cmd.exe
a= 3,45
b= 12,1
3,5*12,1=41,7
Для продолжения нажмите любую клавишу . . .
```

5. запрашивает с клавиатуры два вещественных числа, и выводит на экран результат деления первого числа на второе (вещественные числа выводятся с точностью до 3 знаков после запятой):

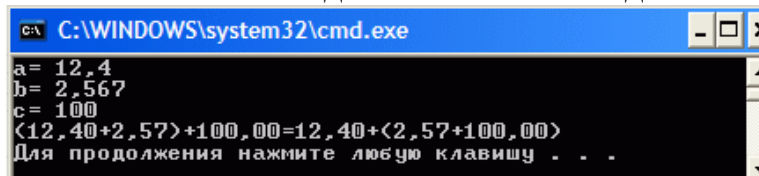


```
C:\WINDOWS\system32\cmd.exe
a= 345,6
b= 12,345
345,600/12,345=27,995
Для продолжения нажмите любую клавишу . . .
```

### Вариант 2.

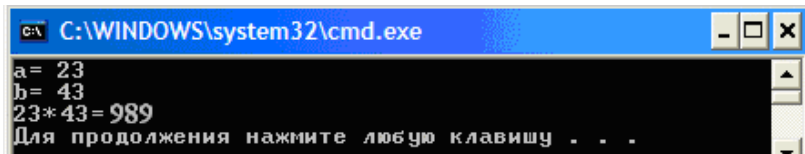
**Задание.** Написать программу, которая реализует диалог с пользователем:

1. запрашивает с клавиатуры три вещественных числа, и выводит на следующее сообщение (вещественные числа выводятся с точностью до 2 знаков после запятой):



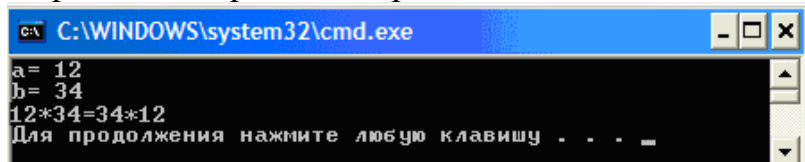
```
C:\WINDOWS\system32\cmd.exe
a= 12,4
b= 2,567
c= 100
(12,40+2,57)+100,00=12,40+(2,57+100,00)
Для продолжения нажмите любую клавишу . . .
```

2. запрашивает с клавиатуры два целых числа, и выводит на экран произведение данных чисел:



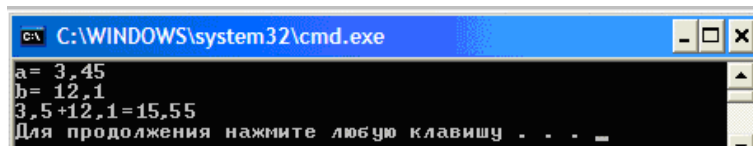
```
C:\WINDOWS\system32\cmd.exe
a= 23
b= 43
23*43= 989
Для продолжения нажмите любую клавишу . . .
```

3. запрашивает с клавиатуры три целых числа, и выводит на экран произведение данных чисел в прямом и обратном порядке:



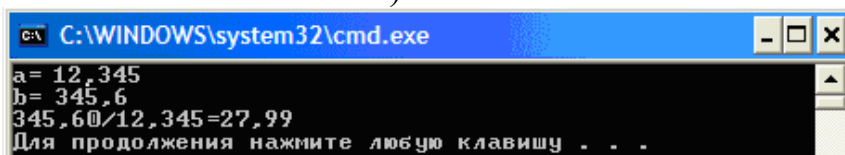
```
C:\WINDOWS\system32\cmd.exe
a= 12
b= 34
12*34=34*12
Для продолжения нажмите любую клавишу . . .
```

4. запрашивает с клавиатуры два вещественных числа, и выводит на экран сумму данных чисел (вещественные числа выводятся с точностью до 2 знаков после запятой):



```
C:\WINDOWS\system32\cmd.exe
a= 3,45
b= 12,1
3,5+12,1=15,55
Для продолжения нажмите любую клавишу . . .
```

5. запрашивает с клавиатуры два вещественных числа, и выводит на экран результат деления второго числа на первое (вещественные числа выводятся с точностью до 2 знаков после запятой):

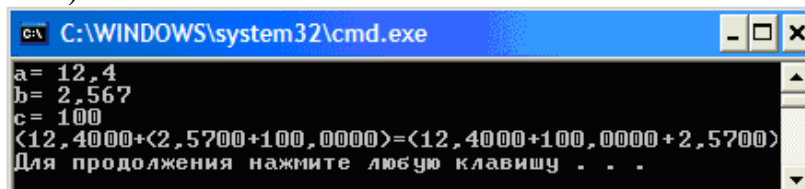


```
C:\WINDOWS\system32\cmd.exe
a= 12,345
b= 345,6
345,60/12,345=27,99
Для продолжения нажмите любую клавишу . . .
```

### Вариант 3.

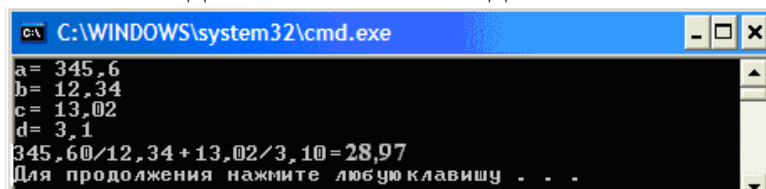
**Задание.** Написать программу, которая реализует диалог с пользователем:

1. запрашивает с клавиатуры три вещественных числа и выводит следующее сообщение  $(a+(b+c))=(a+c+b)$  (вещественные числа выводятся с точностью до 4 знаков после запятой):



```
C:\WINDOWS\system32\cmd.exe
a= 12,4
b= 2,567
c= 100
(12,4000+(2,5700+100,0000))=(12,4000+100,0000+2,5700)
Для продолжения нажмите любую клавишу . . .
```

2. запрашивает с клавиатуры четыре вещественных числа, и выводит на экран результат деления первого числа на второе плюс третьего на четвертое (вещественные числа выводятся с точностью до 2 знаков после запятой):



```
C:\WINDOWS\system32\cmd.exe
a= 345,6
b= 12,34
c= 13,02
d= 3,1
345,60/12,34 + 13,02/3,10=28,97
Для продолжения нажмите любую клавишу . . .
```

3. запрашивает с клавиатуры два целых числа, и выводит на экран результат их суммы, разности и произведения:

```
C:\WINDOWS\system32\cmd.exe
a=15
b=84
a+b=15+84=99    a-b=15-84=-69    a*b=15*84=1260
Для продолжения нажмите любую клавишу . . .
```

4. переменной вещественного типа  $x$  присваивает значение, равное полусумме значений  $a$  и  $b$  с точностью 3 знака после запятой:

```
C:\WINDOWS\system32\cmd.exe
a=3,14
b=51,2
3,14/2+51,2/2=27,170
Для продолжения нажмите любую клавишу . . .
```

5. утраивает значение вещественной переменной  $x$  и выполняет деление на целое число  $y$ , результат округлить до 4 знаков после запятой:

```
C:\WINDOWS\system32\cmd.exe
x=8,265539
y=2
округленный с точностью до 4 знаков результат: 24,796617/2=12,3983
Для продолжения нажмите любую клавишу . . .
```

### Вариант 4.

**Задание.** Написать программу, которая реализует диалог с пользователем:

1. переменной вещественного типа  $x$  присвоить значение на  $\pi$  меньше исходного, результат округлить до 8 знаков после запятой:

```
C:\WINDOWS\system32\cmd.exe
x=6,9830915838
PI=3,14159265358979
x-PI=6,9830915838-3,14159265358979=3,84149893
Для продолжения нажмите любую клавишу . . .
```

2. меняет знак у переменной  $x$  на противоположный и выполняет умножение на вещественное число  $y$ :

```
C:\WINDOWS\system32\cmd.exe
x=2,25
y=6
-x=-2,25    -x*y=-2,25*6=-13,5
Для продолжения нажмите любую клавишу . . .
```

3. запрашивает с клавиатуры три целых числа, и выводит на экран результат модуля разности первого, второго и третьего чисел:

```
C:\WINDOWS\system32\cmd.exe
x=-35
y=4
z=10
k=-49
|x-y-z|=|-35-4-10|=49
Для продолжения нажмите любую клавишу . . .
```

4. переменной целого типа  $x$  присвоить значение, равное половине произведения значений  $a$ ,  $b$ ,  $c$ :

```
C:\WINDOWS\system32\cmd.exe
a=34
b=-2
c=61
x=(a*b*c)/2=(34*-2*61)=-2074
Для продолжения нажмите любую клавишу . . .
```

5. запрашивает с клавиатуры три целых числа, и выводит на экран результат деления первого числа на  $\pi$  (с точностью 4 знака после запятой), сумму данных чисел и результат деления третьего числа на  $E$  (с точностью 3 знака после запятой):

```
C:\WINDOWS\system32\cmd.exe
a=34
b=65
c=24
a/PI=34/3,1416=10,8225   a+b+c=34+65+24=123   c/E=24/2,718=8,829
Для продолжения нажмите любую клавишу . . .
```

### Вариант 5.

**Задание.** Написать программу, которая реализует диалог с пользователем:

1. запрашивает с клавиатуры четыре целых числа, и выводит на экран результат умножения первого числа на третье минус произведение второго числа на четвертое:

```
C:\WINDOWS\system32\cmd.exe
a=9
b=5
c=11
d=3
a*c-b*d=9*11+5*3=84
Для продолжения нажмите любую клавишу . . .
```

2. запрашивает с клавиатуры два вещественных числа, и выводит на экран результат их умножения друг на друга в два столбика: первый столбик – вывод результата с точностью 3 знака после запятой, второй столбик – с точностью 5 знаков после запятой:

```
C:\WINDOWS\system32\cmd.exe
a=5,375864
b=2,4987608
a*b=5,376*2,499=13,433   a*b=5,37586*2,49876=13,43300
Для продолжения нажмите любую клавишу . . .
```

3. переменной вещественного типа  $x$  присваивает значение произведения чисел  $a$ ,  $b$  и  $c$  деленных на их сумму; результат вывести в формате с плавающей и фиксированной точкой с точностью 4 знака после запятой:

```
C:\WINDOWS\system32\cmd.exe
a=34,0987
b=2,125
c=51,284
x=(34,0987*2,125*51,284)/(34,0987+2,125+51,284)=4,246512E+001
x=(34,0987*2,125*51,284)/(34,0987+2,125+51,284)=42,4651
Для продолжения нажмите любую клавишу . . .
```

4. запрашивает с клавиатуры три целых числа и выводит следующее сообщение  $[a*(b+c)]=[(b+c)*a]$ :

```
C:\WINDOWS\system32\cmd.exe
a=35
b=60
c=25
[a*(b+c)]=[(b+c)*a]: [35*(60+25)]=[60+25)*35]=2975
Для продолжения нажмите любую клавишу . . .
```

5. запрашивает с клавиатуры три целых числа, и выводит на экран результат умножения первого числа на PI (с точностью 4 знака после запятой), второго на число E (с точностью 3 знака после запятой) и третьего числа на произведение числа PI на E (с точностью 2 знака после запятой):

```
C:\WINDOWS\system32\cmd.exe
a=5
b=4
c=3
a*PI: 5*3,1416=15,7080   b*E: 4*2,718=10,873   c*PI*E: 3*3,14*2,72=25,62
Для продолжения нажмите любую клавишу . . .
```

### Контрольные вопросы:

1. Дайте определение понятию Операции.
2. Как называется операция увеличения на единицу и уменьшения на единицу?
3. Чем отличаются операции деления «/» и «%»?
4. Приведите формальное определение синтаксиса выражения присваивания.