Программирование на языке Си

Тема 2.Организация ввода-вывода данных. Форматирование.

Что такое переменная?

Переменная – это ячейка в памяти компьютера, которая имеет имя и хранит некоторое значение.

- Значение переменной может меняться во время выполнения программы.
- При записи в ячейку нового значения старое стирается.

Типы переменных

- **int** – целое число (4 байта)
- **double** – вещественное число(8 байт)
- **char** – символ, *character* (2 байта)
- **string** – строка символов (кол-во символов * 2 байта)

Базовые типы C#

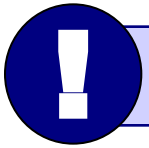
Тип в языке C#	Размер в байтах	Тип .NET	Описание
object		Object	Может хранить все что угодно, так как является всеобщим предком
Логический тип			
bool	1	Boolean	true или false
Целые типы			
sbyte	1	SByte	Целое со знаком (от –128 до 127)
byte	1	Byte	Целое без знака (от 0 до 255)
short	2	Int16	Целое со знаком (от –32 768 до 32 767)
ushort	2	UInt16	Целое без знака (от 0 до 65 535)
int	4	Int32	Целое со знаком (от –2 147 483 648 до 2 147 483 647)
uint	4	UInt	Целое число без знака (от 0 до 4 294 967 295)
long	8	Int64	Целое со знаком (от –9 223 372 036 854 775 808 до 9 223 372 036 854 775 807)
ulong	8	UInt64	Целое без знака (от 0 до 0ffffffffffffff)

Тип в языке C#	Размер в байтах	Тип .NET	Описание
Вещественные типы			
float	4	Single	Число с плавающей точкой двойной точности. Содержит значения приблизительно от $\pm 1{,}5 \cdot 10^{-45}$ до $\pm 3{,}4 \cdot 10^{38}$ с 7 значащими цифрами
double	8	Double	Число с плавающей точкой двойной точности. Содержит значения приблизительно от $\pm 5{,}0 \cdot 10^{-324}$ до $\pm 1{,}7 \cdot 10^{308}$ с 15—16 значащими цифрами
Символьный тип			
char	2	Char	Символ Unicode
Строковый тип			
string		String	Строка из Unicode-символов
Финансовый тип			
decimal	12	Decimal	Число до 28 знаков с фиксированным положением десятичной точки. Обычно используется в финансовых расчетах.

Имена переменных

Могут включать

- латинские буквы (A-Z, a-z)
- знак подчеркивания _
- цифры 0-9



Имя не может начинаться с цифры!

НЕ могут включать

- русские буквы
- пробелы
- скобки, знаки +, =, !, ? и др.

Какие имена правильные?

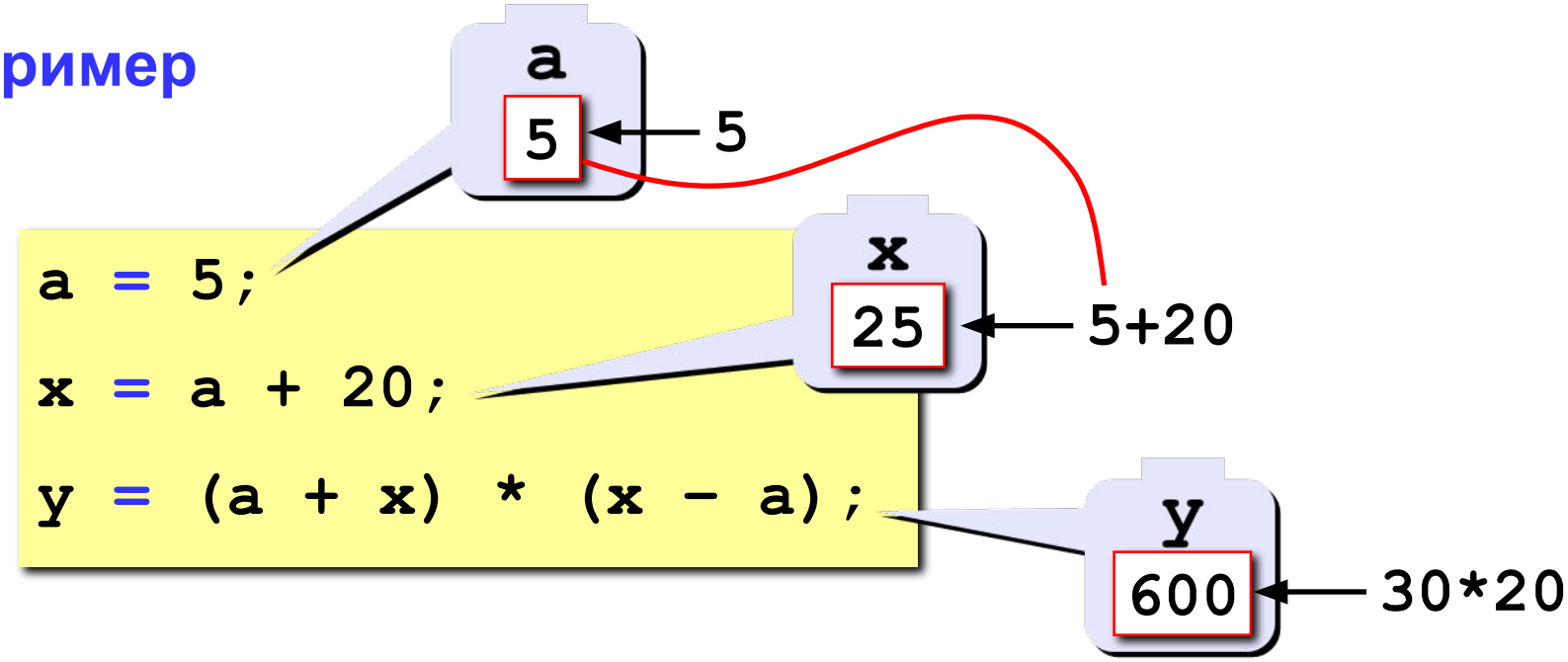
AXby R&B 4Wheel Вася “PesBarbos”
TU154 [QuQu] _ABBA A+B

Оператор присваивания

Оператор – это команда языка программирования высокого уровня.

Оператор присваивания служит для изменения значения переменной.

Пример



Объявление переменных

Объявить переменную = определить ее имя, тип, начальное значение, и выделить ей место в памяти.

```
main()
{
int a;
double x, y, z;
int Tu104, Il86=23, Yak42;
double x=4.56, y, z;
char c, c2='A', m;
}
```

целая переменная **a**

вещественные переменные **x**, **y** и **z**

целая и дробная части отделяются **точкой**

целые переменные **Tu104**, **Il86** и **Yak42**

вещественные переменные **x**, **y** и **z**
x = 4,56

символьные переменные **c**, **c2** и **m**
c2 = 'A'

Если начальное значение не задано, в этой ячейке находится «мусор»!

Объявление и инициализация переменной

```
using System;
namespace MyProject
{
   class Program
   {
      static void Main()
      {

         int i=10;                    // объявление и инициализация
                                      // целочисленной переменной i
         Console.WriteLine(i);        // просмотр значения переменной
         i=100;                       // изменение значение переменной
         Console.WriteLine(i);        // вывод значения i на экран
      }
   }
}
```

```
Console.ReadKey();
```

Объявление и инициализация переменной

```
static void Main()
{
   int i;   // объявление переменной без инициализации
   Console.WriteLine(i);   // просмотр значения переменной
}
```

Write **или** **WriteLine**

```
Console.Write("++++++++++++++++++++++++++++");
Console.Write("9999999999999");
Console.ReadKey();
```

Операторы ввода и вывода данных

`Convert.ToInt32(Console.ReadLine());`
– ввод данных

`Console.WriteLine(……);`
– вывод данных

Console.WriteLine(x);	на экран выводится значение идентификатора x
Console.WriteLine("x=" + x +"y=" + y);	на экран выводится строка, образованная последовательным слиянием строки "x=", значения x, строки "y=" и значения y
Console.WriteLine ("x={0} y={1}", x, y);	на экран выводится строка, формат которой задан первым аргументом метода, при этом вместо параметра {0} выводится значение x, а вместо {1} – значение y

```
// На экран выводится значение идентификатора x
Console.WriteLine(x);
```

Пример:

```
x=5;
Console.WriteLine(x);
```

```
/* На экран выводится строка, образованная последовательным
слиянием строки "x=", значения x, строки "y=" и значения y */
Console.WriteLine("x=" + x +"y=" + y);
```

Пример:

```
x=5;
y=6;
Console.WriteLine(“x=“+x+”y=“+y);
```

x=5y=6

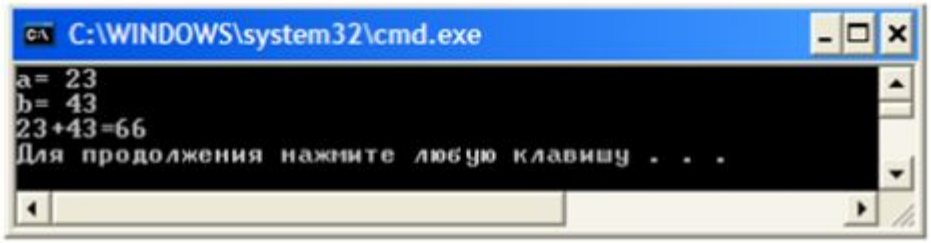
Пример :

```
Console.Write(“a=“);
a =Convert.ToInt32(Console.ReadLine());
```

a=

Практикум _2 к лекции 3

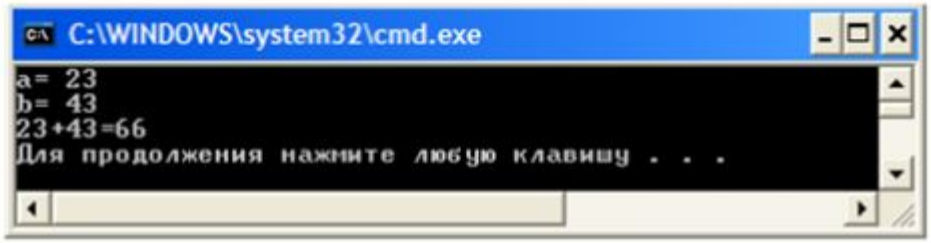
1) запрашивает с клавиатуры два целых числа и выводит на экран сумму данных чисел:



Пример 1:

```
Int a=23, b=43, c;
Console.WriteLine("a="+a);
Console.WriteLine("b="+b);
c=a+b;
Console.WriteLine(a+"+"+b+"="+c);
```

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



Пример:

```
Int a, b, c;
Console.Write("a=");
a=Convert.ToInt32(Console.ReadLine());

Console.WriteLine("b=");
b=Convert.ToInt32(Console.ReadLine());
c=a+b;
Console.WriteLine(a+"+"+b+"="+c);
```

Оператор присваивания

Общая структура:

куда записать

что

```
имя переменной = выражение;
```

Арифметическое выражение может включать

- константы (постоянные)
- имена переменных
- знаки арифметических операций:

```
+ -   *   /   (*0.1)
```

умножение

деление

Если деление с дробями

- вызовы функций
- круглые скобки ( )

Для чего служат круглые скобки?

Математические операции

/ целочисленное деление

% остаток от деления

Математические функции

Название	Описание
Math.Abs(<выражение>)	Модуль
Math.Ceiling(<выражение>)	Округление для большего целого
Math.Cos(<выражение>)	Косинус
Math.E	Число e
Math.Exp(<выражение>)	Экспонента
Math.Floor(<выражение>)	Округление до меньшего целого
Math.Log(<выражение>)	Натуральный логарифм
Math.Log10(<выражение>)	Десятичный логарифм
Math.Max(<выражение1>, <выражение2>)	Максимум из двух значений
Math.Min(<выражение1>, <выражение2>)	Минимум из двух значений
Math.PI	Число π
Math.Pow(<выражение1>, <выражение2>)	Возведение в степень
Math.Round(<выражение>)	Простое округление
Math.Sign(<выражение>)	Знак числа
Math.Sin(<выражение>)	Синус
Math.Sqrt(<выражение>)	Квадратный корень
Math.Tan(<выражение>)	Тангенс

Особенность деления в Си

При делении целых чисел остаток отбрасывается!

```
{
int a = 7;
double x;
x = a / 4;
x = 4 / a;
x = double(a) / 4;
x = 1.0*a / 4;
}
```

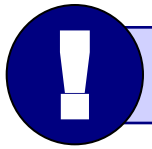
1

0

1.75

1.75

Особенность деления в Си



При делении целых чисел остаток отбрасывается!

Для задач, где требуется повышенная точность, и задач с автоматической проверкой лучше всего использовать тип данных:

Decimal

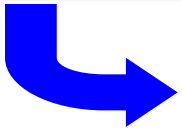
Порядок выполнения операций

1) вычисление выражений в скобках
2) умножение, деление и %(остаток от деления) слева направо
3) сложение и вычитание слева направо

```
      1  2   5   3   6   4
z = (5*a+c)/(a*b)*(b-c);
```

$$z = \frac{5a + c}{ab}(b - c)$$

```
      1  2   4  5   3   6
z = (5*a+c)/a*(b-c)/ b;
```

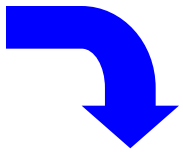


$$z = \frac{5a + c}{ab}(b - c)$$

Порядок выполнения операций

1) вычисление выражений в скобках
2) умножение, деление и %(остаток от деления) слева направо
3) сложение и вычитание слева направо

$$x = \frac{5c^2 - d(a+b)}{(c+d)(d-2a)}$$



2 3 5 4 1 10 6 9 8 7

```
x =(5*c*c-d*(a+b))/((c+d)*(d-2*a))
```

Задача. Дано два числа ***a*** и ***b***. Найдите гипотенузу треугольника с заданными катетами.

1. **Ввести число a**
2. **Ввести число b**
3. **Вычислить число c по формуле**
4. **Вывести c**

Ввод данных с клавиатуры

```
        oid Main(string[] args)
        {
            int a, b;
            decimal c;
            a = Convert.ToInt32(Console.ReadLine());
            b = Convert.ToInt32(Console.ReadLine());
            c = Math.Sqrt(a * a + b * b);

            Console.WriteLine(c);
            Console.ReadKey();
        }
}
```

Подключение «математики»

Вывод данных

Практикум _1 к лекции 3

Использование управляющих последовательностей

Управляющей последовательностью называют определенный символ, предваряемый обратной косой чертой.

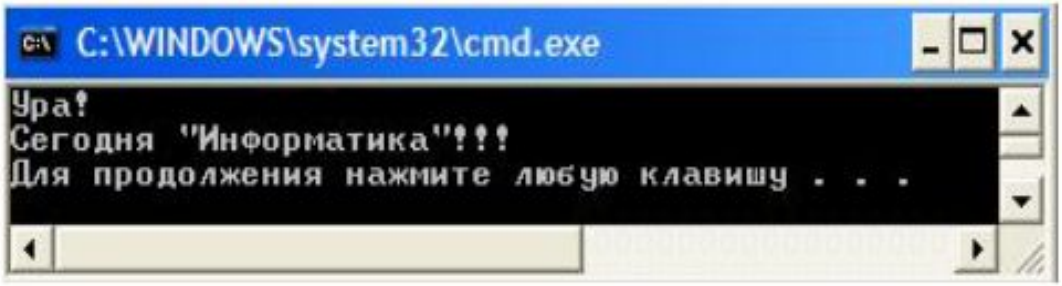
Используется для представления кодов символов , не имеющих графического представления (перевод курсора и др.) или символов, имеющих специальное обозначение.

Управляющие символы

Вид	Наименование
\a	Звуковой сигнал
\b	Возврат на шаг назад
\f	Перевод страницы
\n	Перевод строки
\r	Возврат каретки
\t	Горизонтальная табуляция
\v	Вертикальная табуляция
\\	Обратная косая черта
\'	Апостроф
\"	Кавычки

Пример:

```
static void Main()
{
  Console.WriteLine("Ура!\nСегодня \"Информатика\"!!!");
}
```



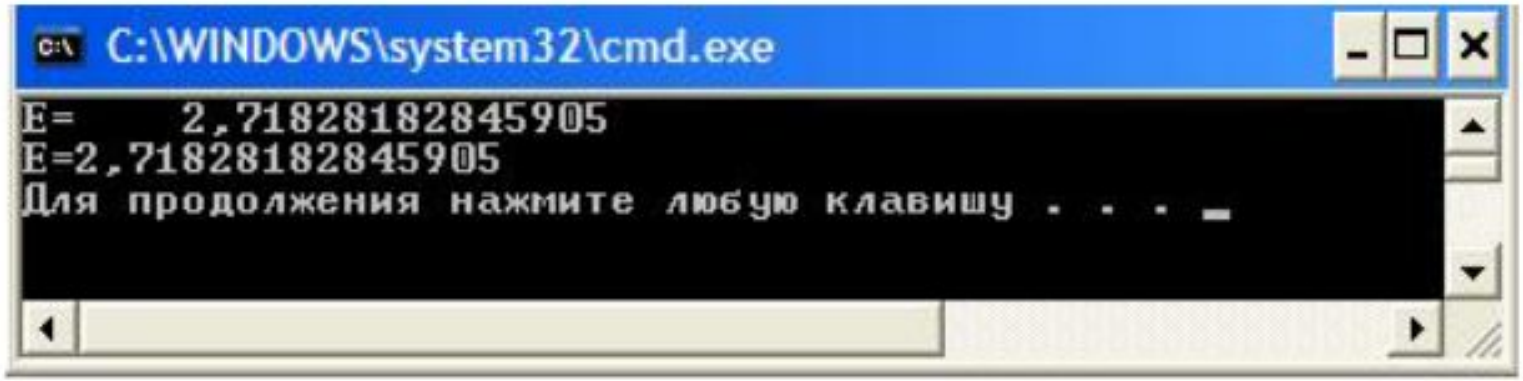
Замечание. Вместо управляющей последовательности \n можно использовать константу Enviropment.NewLine. Она более универсальна, так как ее значение зависит от контекста и операционной системы, в которой запускается программа.

Задание. Измените программу так, чтобы все сообщение выводилось в одну строку, а после вывода сообщения раздавался звуковой сигнал.

Управление размером поля вывода

Первым аргументом указывается строка вида {n,m} - где n определяет номер идентификатора из списка, а m- количество позиций (размер поля вывода), отводимых под значение данного идентификатора.

```
static void Main()
{
   double x = Math.E;
   Console.WriteLine("E={0,20}", x);
   Console.WriteLine("E={0,10}", x);
}
```

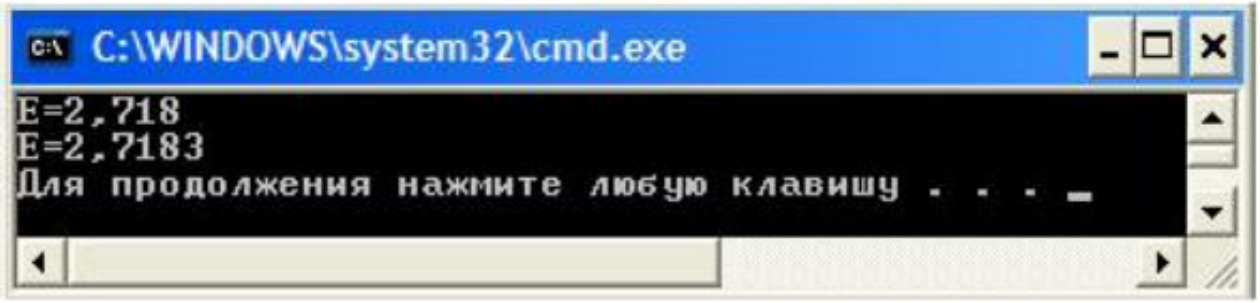


Управление размещением вещественных чисел

Первым аргументом указывается строка вида {n:##.###} – где n определяет номер идентификатора, а ##.### определяет формат вывода вещественного числа.

Целая часть две позиции, дробная – три.

```
static void Main()
{
   double x= Math.E;
   Console.WriteLine("E={0:##.###}", x);
   Console.WriteLine("E={0:.####}", x);
}
```

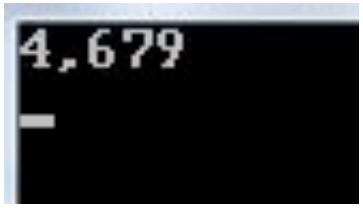


Задание. Измените программу так, чтобы число e выводилось на экран с точностью до 6 знаков после запятой.

Примеры:

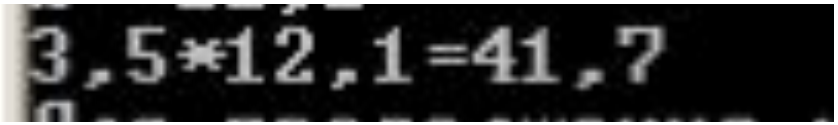
a = 4.678999;

Console.WriteLine("{0:##.###}", a);



Console.Write("{0:##.#}", a);
Console.Write(“ * ”);
Console.Write("{0:##.#}", b);

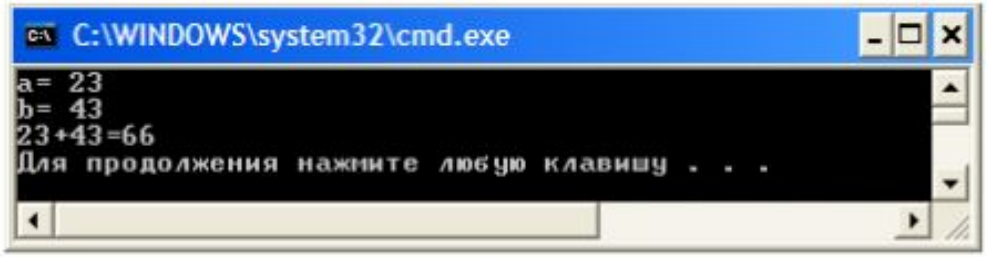
Console.Write(“=“ + c);



Практикум

I. Написать программу, которая реализует диалог с пользователем:

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




```
Int a,b,c;
Console.Write(“a=“);
a =Convert.ToInt32(Console.ReadLine());
……..
c=a+b;
Console.WriteLine(a+“+“+b+”=“+c);
```

Практикум _2 к лекции 3