

# ОСНОВЫ ПРОГРАММИРОВАНИЯ

Самойлов Михаил Юрьевич

# Операции над числами

В языке C# существует большой набор математических операций для преобразования числовых значений. В языке C# существует специальная библиотека, которая содержит большой набор функций для работы с числами.

# Math

- константы -  $\pi$ ,  $E$ ;
- тригонометрические функции -  $\sin$ ,  $\cos$ ,  $\tan$ ;
- обратные тригонометрические функции -  $\arcsin$ ,  $\arccos$ ,  $\arctan$ ;
- гиперболические функции -  $\tanh$ ,  $\sinh$ ,  $\cosh$ ;
- экспоненту и логарифмические функции -  $\exp$ ,  $\log$ ,  $\log_{10}$ ;
- модуль, корень, знак -  $\text{abs}$ ,  $\text{sqrt}$ ,  $\text{sign}$ ;
- функции округления -  $\text{ceil}$ ,  $\text{floor}$ ,  $\text{round}$ ;
- минимум, максимум, степень, остаток -  $\text{min}$ ,  $\text{max}$ ,  $\text{pow}$ .

# Константы - $\pi$ , $e$

- $\pi$  - 3,14159265358979
- $e$  - 2,71828182845905

# Константы - PI, E

```
double p = Math.PI;  
double e = Math.E;  
Console.WriteLine(p); // 3,14159265358979  
Console.WriteLine(e); // 2,71828182845905  
Console.ReadLine();
```

# Тригонометрические функции - Sin, Cos, Tan

- `Cos(double d)`: возвращает косинус угла  $d$
- `Sin(double d)`: возвращает синус угла  $d$
- `Tan(double d)`: возвращает тангенс угла  $d$

# Тригонометрические функции - Sin, Cos, Tan

```
double p = Math.PI / 6;  
double s = Math.Sin(p);  
double c = Math.Cos(p);  
double t = Math.Tan(p);  
Console.WriteLine(s); // 0,5  
Console.WriteLine(c); // 0,866025403784439  
Console.WriteLine(t); // 0,577350269189626  
Console.ReadLine();
```

# Обратные тригонометрические функции - ASin, ACos, ATan

- `Acos(double value)`: возвращает арккосинус `value`. Параметр `value` должен иметь значение от -1 до 1
- `Asin(double value)`: возвращает арксинус `value`. Параметр `value` должен иметь значение от -1 до 1
- `Atan(double value)`: возвращает арктангенс `value`

# Обратные тригонометрические функции - ASin, ACos, ATan

```
double arcs = Math.Asin(1);  
double arcc = Math.Acos(1);  
double arct = Math.Atan(1);  
Console.WriteLine(arcs); // 1,5707963267949  
Console.WriteLine(arcc); // 0  
Console.WriteLine(arct); // 0,785398163397448  
Console.ReadLine();
```

# Гиперболические функции - Tanh, Sinh, Cosh

- `Sinh(double value)`: возвращает гиперболический синус угла `value`
- `Tanh(double value)`: возвращает гиперболический тангенс угла `value`
- `Cosh(double value)`: возвращает гиперболический косинус угла `value`

# Гиперболические функции - Tanh, Sinh, Cosh

```
double sh = Math.Cosh(1);  
double ch = Math.Sinh(1);  
double th = Math.Tanh(1);  
Console.WriteLine(sh); // 1,54308063481524  
Console.WriteLine(ch); // 1,1752011936438  
Console.WriteLine(th); // 0,761594155955765  
Console.ReadLine();
```

# Экспонента и логарифмические функции - Exp, Log, Log10

- `Exp(double d)`: возвращает основание натурального логарифма, возведенное в степень `d`
- `Log(double d)`: возвращает натуральный логарифм числа `d`
- `Log(double d, double newBase)`: возвращает логарифм числа `d` по основанию `newBase`
- `Log10(double d)`: возвращает десятичный логарифм числа `d`

# Экспонента и логарифмические функции - Exp, Log, Log10

```
double e = Math.E;
double exp = Math.Exp(1);
double log = Math.Log(e);
double log2 = Math.Log(4, 2);
double log10 = Math.Log10(1000);
Console.WriteLine(exp); // 2,71828182845905
Console.WriteLine(log); // 1
Console.WriteLine(log2); // 2
Console.WriteLine(log10); // 3
Console.ReadLine();
```

# Модуль, корень, знак - Abs, Sqrt, Sign

- `Abs(double value)`: возвращает абсолютное значение для аргумента `value`
- `Sqrt(double value)`: возвращает квадратный корень числа `value`
- `Sign(double value)`: возвращает число 1, если число `value` положительное, и -1, если значение `value` отрицательное. Если `value` равно 0, то возвращает 0

# Модуль, корень, знак - Abs, Sqrt, Sign

```
double abs = Math.Abs(-12);  
double sqrt = Math.Sqrt(9);  
double sign = Math.Sign(-7);  
Console.WriteLine(abs); // 12  
Console.WriteLine(sqrt); // 3  
Console.WriteLine(sign); // -1  
Console.ReadLine();
```

# Функции округления - Ceiling, Floor, Round

- `Ceiling(double value)`: возвращает наименьшее целое число с плавающей точкой, которое не меньше `value`
- `Floor(decimal d)`: возвращает наибольшее целое число, которое не больше `d`
- `Round(double d)`: возвращает число `d`, округленное до ближайшего целого числа

# Функции округления - Ceiling, Floor, Round

```
double c = Math.Ceiling(2.47);  
double r = Math.Round(9.5);  
double f = Math.Floor(3.7);  
Console.WriteLine(c); // 3  
Console.WriteLine(r); // 10  
Console.WriteLine(f); // 3  
Console.ReadLine();
```

# Минимум, максимум, степень, остаток

## - Min, Max, Pow

- `Min(double a, double b)`: возвращает минимальное число из `a` и `b`
- `Max(double a, double b)`: возвращает максимальное число из `a` и `b`
- `Pow(double a, double b)`: возвращает число `a`, возведенное в степень `b`

# Минимум, максимум, степень, остаток - Min, Max, Pow

```
double min = Math.Min(1, 5);  
double max = Math.Max(1, 5);  
double pow = Math.Pow(3, 2);  
Console.WriteLine(min); // 1  
Console.WriteLine(max); // 5  
Console.WriteLine(pow); // 9  
Console.ReadLine();
```

# Случайные числа

Генерация случайных чисел требуется во многих приложениях. Часто бывает так что в программе используется некоторый элемент случайности. В C# для этого используется случайный класс Random.

# Класс Random

Для генерации случайных чисел нужно создать переменную данного класса. Делается это несколько особенным способом:

```
Random rn = new Random();
```

Здесь создана переменная с именем `rn`. С помощью нее и можно будет в дальнейшем генерировать случайные числа.

# Генерация целых чисел.

Для генерации целых чисел используется команда `Next`, которая имеет несколько вариаций:

- `Next()` - метод без параметров выдает целые положительные числа во всем положительном диапазоне типа `int`;
- `Next(int max)` - выдает целые положительные числа в диапазоне `[0, max)`;
- `Next(int min, int max)` - выдает целые положительные числа в диапазоне `[min, max)`.

# Генерация целых чисел.

```
Random rn = new Random();  
int r1 = rn.Next();  
int r2 = rn.Next(20);  
int r3 = rn.Next(25, 40);  
Console.WriteLine(r1); // 2080124162  
Console.WriteLine(r2); // 13  
Console.WriteLine(r3); // 26  
Console.ReadLine();
```

# Генерация вещественных чисел.

Для генерации вещественных чисел можно использовать ту же команду Next, с использованием некоторых дополнительных арифметических операций:

```
Random rn = new Random();  
double d = rn.Next(100) / 100.0;  
Console.WriteLine(d); // 0,34  
Console.ReadLine();
```

# Генерация вещественных чисел.

Так же для генерации случайных вещественных чисел используется специальную, предназначенную для этого команду:

- `NextDouble()` - выдает вещественные положительные числа в диапазоне  $[0, 1)$ ;

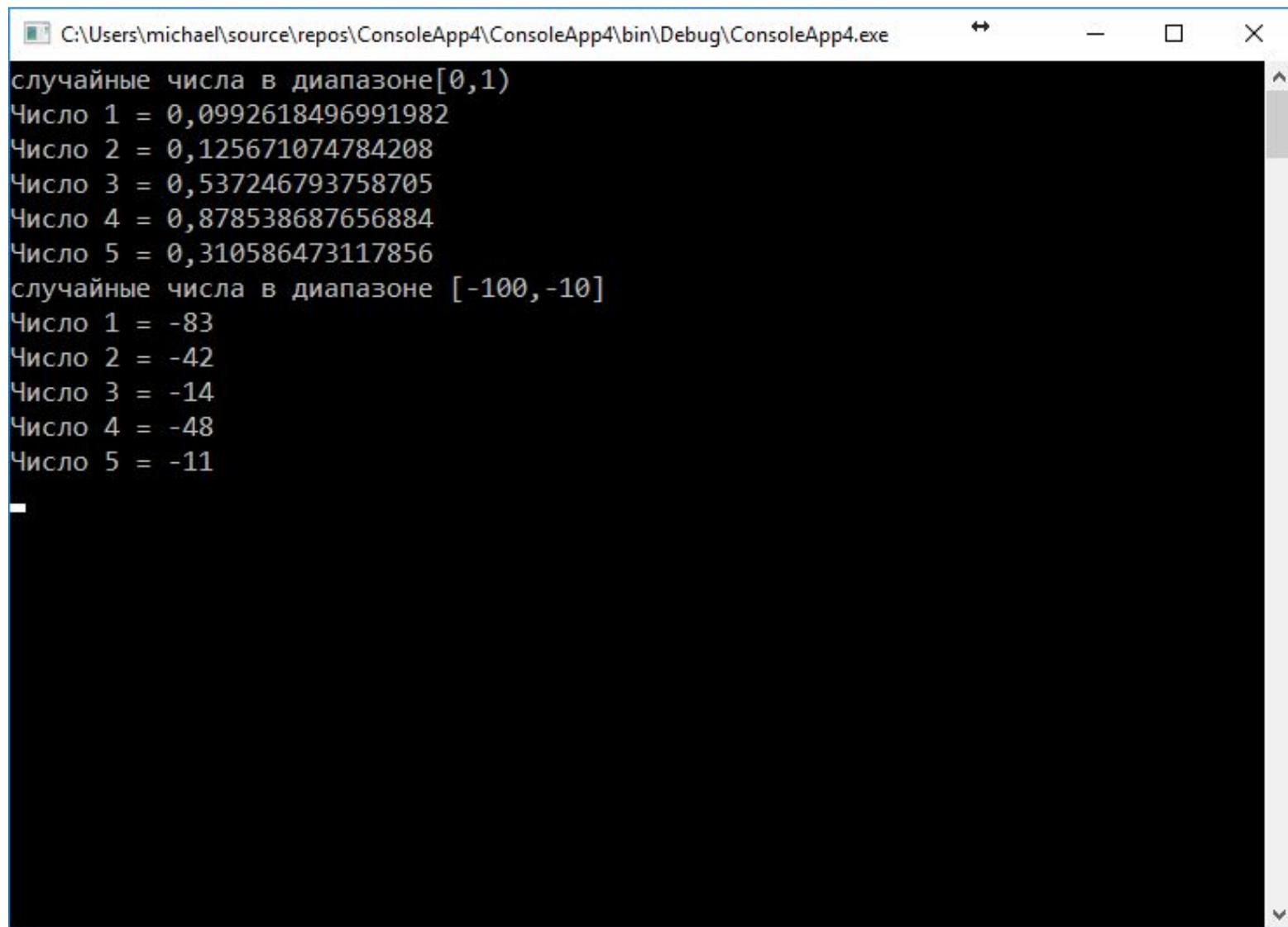
# Генерация вещественных чисел.

```
Random rn = new Random();  
double d = rn.NextDouble();  
Console.WriteLine(d); // 0,279956102967242  
Console.ReadLine();
```

# Пример

```
Random rn = new Random();
Console.WriteLine("случайные числа в диапазоне[0,1)");
for (int i = 1; i <= 5; i++)
{
    Console.WriteLine("Число " + i + " = " + rn.NextDouble());
}
int min = -100, max = -10;
Console.WriteLine("случайные числа в диапазоне [" + min + ", " + max + "]);
for (int i = 1; i <= 5; i++)
{
    Console.WriteLine("Число " + i + " = " + rn.Next(min, max));
}
Console.ReadLine();
```

# Пример



The image shows a screenshot of a Windows console window. The title bar at the top reads "C:\Users\michael\source\repos\ConsoleApp4\ConsoleApp4\bin\Debug\ConsoleApp4.exe". The console output is as follows:

```
случайные числа в диапазоне[0,1)
Число 1 = 0,0992618496991982
Число 2 = 0,125671074784208
Число 3 = 0,537246793758705
Число 4 = 0,878538687656884
Число 5 = 0,310586473117856
случайные числа в диапазоне [-100,-10]
Число 1 = -83
Число 2 = -42
Число 3 = -14
Число 4 = -48
Число 5 = -11
```

The console window has a black background with white text. The output is organized into two sections: the first section shows five random numbers in the range [0,1), and the second section shows five random numbers in the range [-100,-10]. The window includes standard Windows window controls (minimize, maximize, close) in the top right corner.

# Задачи

**Задача 1.** Считать с клавиатуры 2 катета прямоугольного треугольника, найти гипотенузу и вывести на экран.

**Задача 2.** Посчитать сумму корней чисел от 1 до  $N$ . Число  $N$  вводится с клавиатуры.

**Задача 3.** Сгенерировать  $N$  случайных целых чисел от 1 до  $N$ . Вывести данные числа на экран. Число  $N$  вводится с клавиатуры.