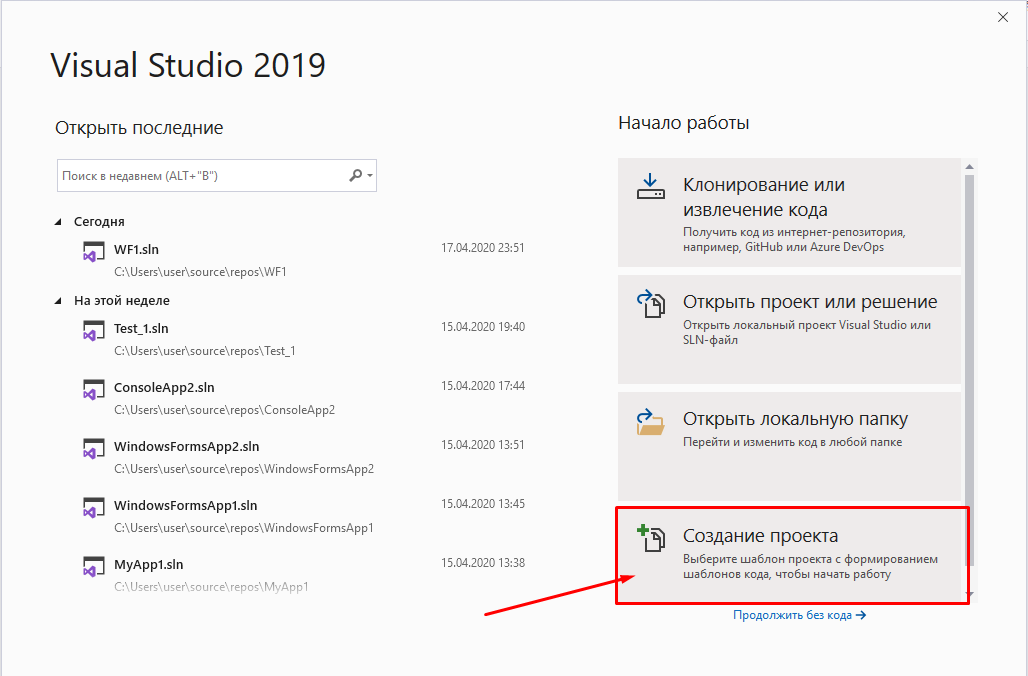
Переменные и типы данных

Цель:

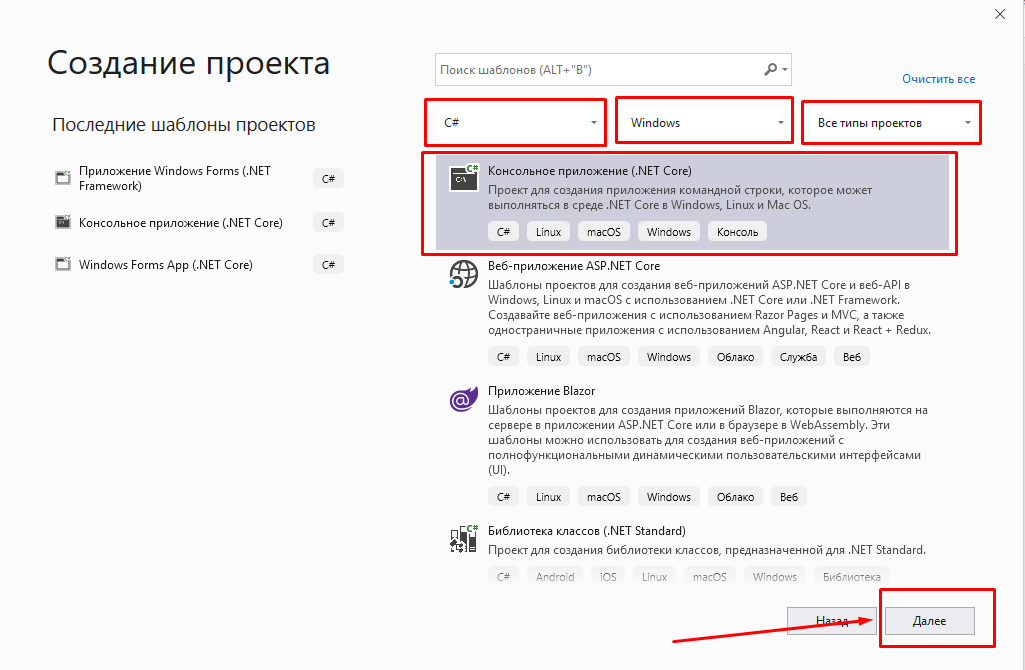
* рассмотрение понятия переменной, константы и типа данных;
* рассмотрение арифметических операторов и операторов сравнения.

Порядок выполнения работы:

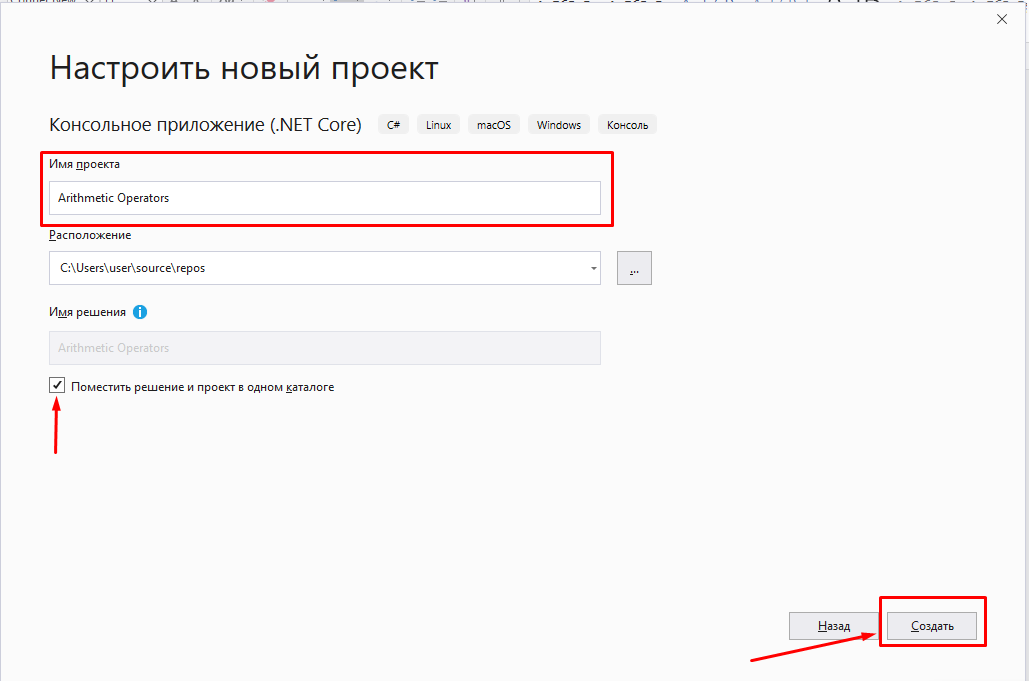
1. Создайте консольное приложение с именем Arithmetic Operators. Для этого следуйте указаниям на рисунках ниже:



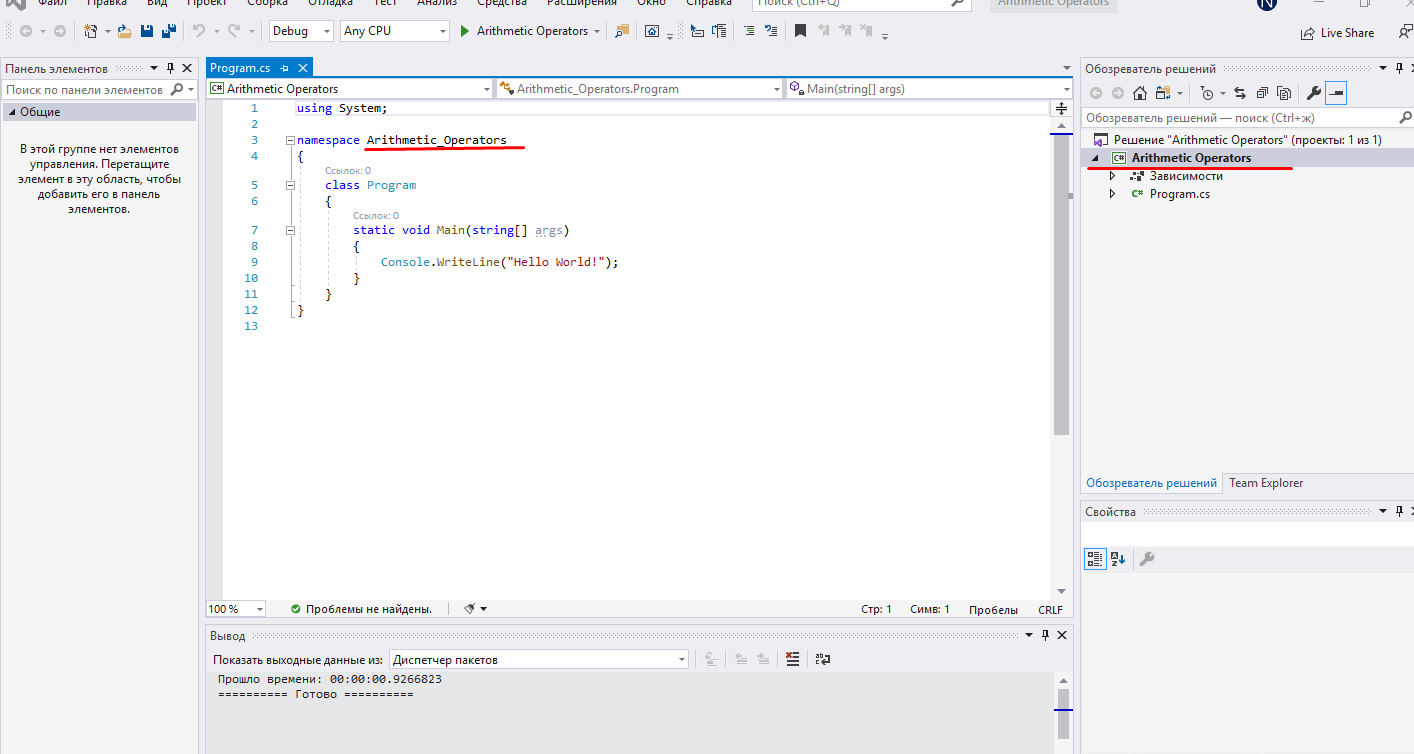
Далее выберите тип проекта:



Введите название:



Обратите внимание, что проект и пространство имен называются согласно введенного имени. Не изменяйте этих имен, понятие пространства имен мы изучим далее:



Далее введите код, обратите внимание на стандартную структуру программы, менять структуру нельзя, скобки тоже удалять нельзя.

using System;

// Арифметические операторы (Arithmetic Operators) - +, −, \*, /, %

namespace Arithmetic\_Operators // Здесь будет указано имя вашего проекта, не меняйте его

{

class Program

{

static void Main(string[] args)

{

// Сумма (+)

byte summand1 = 1, summand2 = 2; // Множественное объявление.

int sum = 0;

sum = summand1 + summand2;

Console.WriteLine(sum);

// Разность (-)

byte minuend = 5, subtrahend = 3;

int difference = 0;

difference = minuend - subtrahend;

Console.WriteLine(difference);

// Произведение (\*)

byte factor1 = 2, factor2 = 3;

int product = 0;

product = factor1 \* factor2;

Console.WriteLine(product);

// Деление (/)

byte dividend = 5, divisor = 2;

int quotient = 0, remainder = 0;// remainder - остаток от деления

quotient = dividend / divisor;

Console.WriteLine(quotient);

// Остаток от деления (%)

remainder = dividend % divisor;

Console.WriteLine(remainder);

// Задержка

Console.ReadKey();

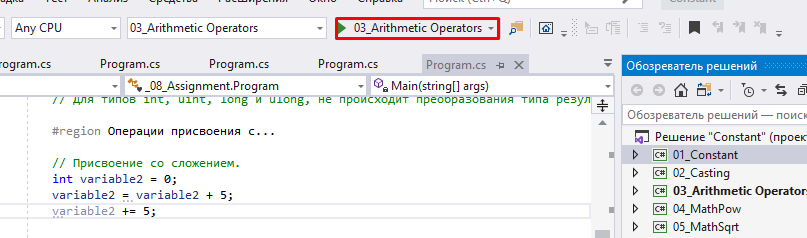
}

}

}

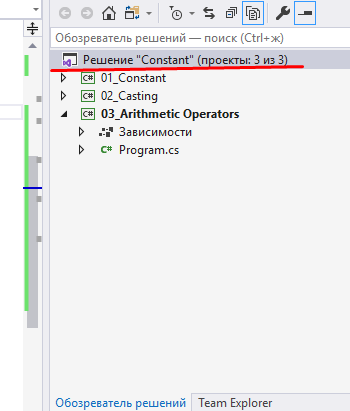
И запустите проект на выполнение.

Для этого нажмите:

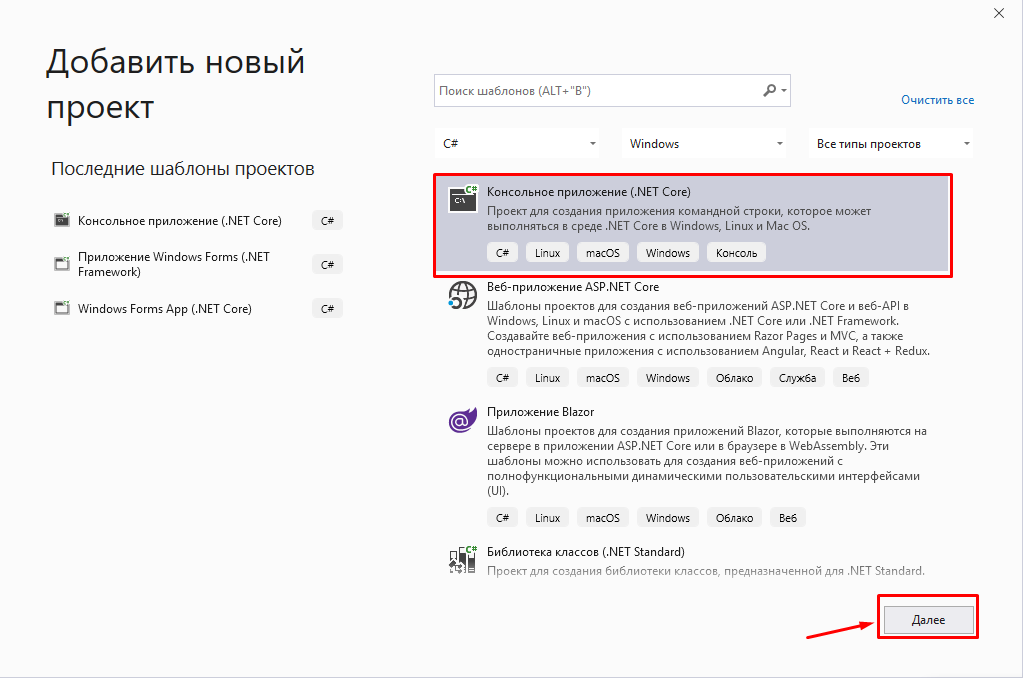


Попробуйте изменить значение переменных и проверьте работу программы.

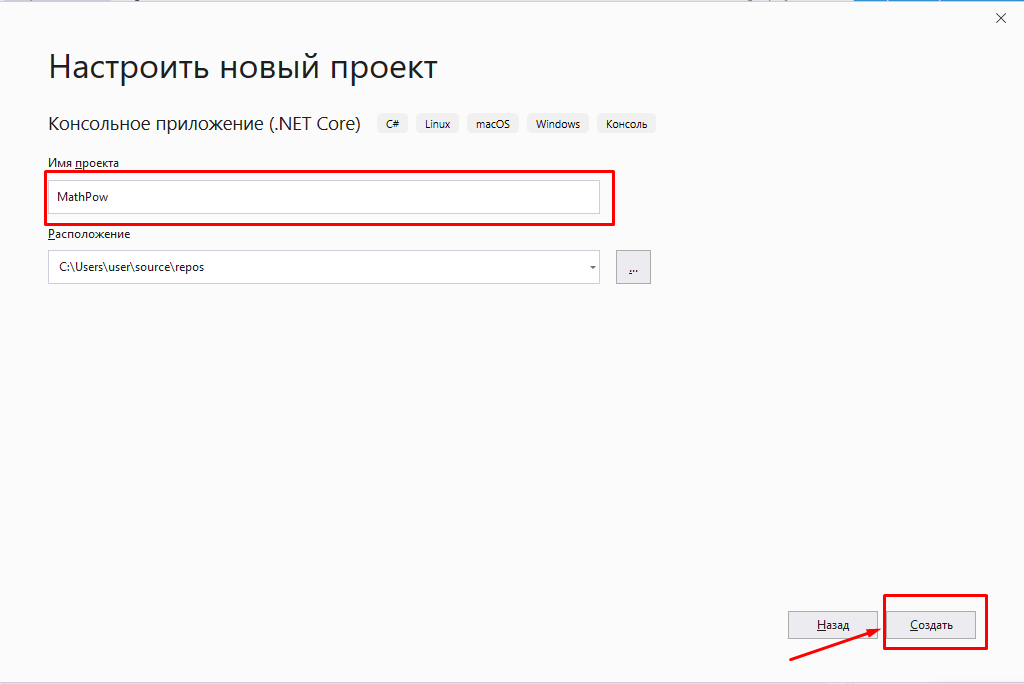
В этом же решении создадим еще один проект «MathPow». Для этого выделить имя решения в Обозревателе решений:



Далее по нажатию правой клавиши выбрать: «Добавить», затем «Создать проект».



Далее ввести имя проекта:



В этом проекте рассмотрим возведение в степень.

using System;

/\* Math.Pow() - возведение числа в степень (1-ый аргумент - число, которое возводим в степень,

2-ой – степень, в которую возводим число)\*/

namespace MathPow

{

class Program

{

static void Main(string[] args)

{

double x = 2, y = 8;

double result = Math.Pow(x, y);

Console.WriteLine(result);

// Задержка

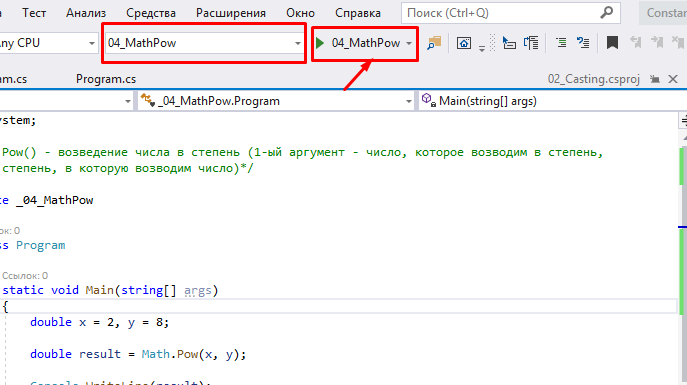
Console.ReadKey();

}

}

}

Для запуска второго проекта в этом решении необходимо выбрать проект для запуска:



После проверки результата создайте еще один проект (аналогично предыдущему примеру) с названием MathSqrt

В созданном проекте проверьте работоспособность кода:

using System;

// Math.Sqrt() - математическая функция, которая извлекает квадратный корень

namespace MathSqrt

{

class Program

{

static void Main(string[] args)

{

double x = 256;

double result = Math.Sqrt(x);

Console.Write("Квадратный корень равен: ");

Console.WriteLine(result);

// Задержка

Console.ReadKey();

}

}

}

Запустите проект.

Далее создайте в этом же решении проект с именем Comparsion. Это проект будет посвящен работе с логическими выражениями:

using System;

// Операции сравнения и проверки на равенство (<, <=, >, >=, ==, !=)

namespace Comparison

{

class Program

{

static void Main(string[] args)

{

byte value1 = 0, value2 = 1;

bool result = false;

// Меньше

result = value1 < value2;

Console.WriteLine(result);

// Больше

result = value1 > value2;

Console.WriteLine(result);

// Меньше либо равно

result = value1 <= value2;

Console.WriteLine(result);

// Больше либо равно

result = value1 >= value2;

Console.WriteLine(result);

// Равно

result = value1 == value2;

Console.WriteLine(result);

// Не равно

result = value1 != value2;

Console.WriteLine(result);

// Задержка

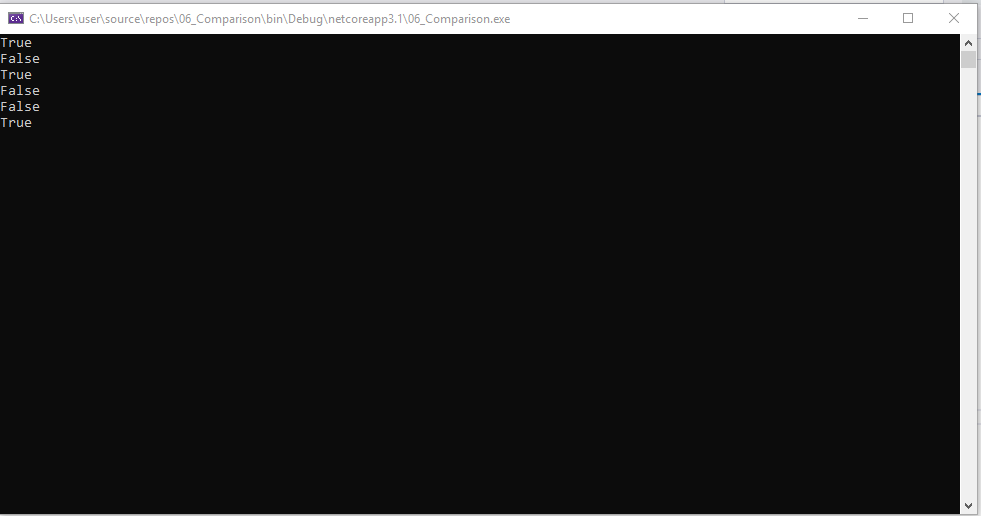
Console.ReadKey();

}

}

}

Проверьте работоспособность кода:



Далее создайте в этом же решении проект IncDec

using System;

// Операторы Инкремента и Декремента (Increment and Decrement Operators)

namespace IncDec

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("----- Постфиксный инкремент"); // Post-increment

byte number1 = 0;

Console.WriteLine(number1++); // Сначала выводим на экран, потом увеличиваем на 1.

Console.WriteLine(number1);

Console.WriteLine("----- Префиксный инкремент"); // Pre-increment

byte number2 = 0;

Console.WriteLine(++number2); // Сначала увеличиваем на 1, потом выводим на экран.

Console.WriteLine("----- Постфиксный декремент"); // Post-decrement

sbyte number3 = 0;

Console.WriteLine(number3--); // Сначала выводим на экран, потом уменьшаем на 1.

Console.WriteLine(number3);

Console.WriteLine("----- Префиксный декремент"); // Pre-decrement

sbyte number4 = 0;

Console.WriteLine(--number4); // Сначала уменьшаем на 1, потом выводим на экран.

// Задержка

Console.ReadKey();

}

}

}

Проверьте работоспособность программного кода.

Создайте проект с именем Assignment и добавьте операторы вывода всех переменных Console.WriteLine.

using System;

namespace Assignment

{

class Program

{

static void Main(string[] args)

{

// ПРАВИЛО:

// Все арифметические операции производимые над двумя значениями типа (byte, sbyte, short, ushort)

// в качестве результата, возвращают значение типа int.

// Присвоение со сложением для типа byte.

byte variable1 = 0;

//variable1 = variable1 + 5; // ОШИБКА: Попытка неявного преобразования значения результата, тип int в тип byte.

//variable1 = (byte)variable1 + 5; // ОШИБКА: Происходит преобразование типа byte в тип byte, раньше выполнения операции сложения.

variable1 = (byte)(variable1 + 5); // Громоздкое решение.

variable1 += 5; // Элегантное решение.

//variable1 += 5000; // Ошибка. т.к. значение правой части выражения не должно превышать диапазон допустимых значений типа переменной

// ПРАВИЛО:

// Для типов int, uint, long и ulong, не происходит преобразования типа результата арифметических операций.

#region Операции присвоения с...

// Присвоение со сложением.

int variable2 = 0;

variable2 = variable2 + 5;

variable2 += 5;

// Присвоение с вычитанием.

uint variable3 = 0;

variable3 = variable3 - 5;

variable3 -= 5;

// Присвоение с умножением.

long variable4 = 0;

variable4 = variable4 \* 5;

variable4 \*= 5;

// Присвоение с делением.

ulong variable5 = 0;

variable5 = variable5 / 5;

variable5 /= 5;

// Присвоение остатка от деления.

long variable6 = 0;

variable6 = variable6 % 5;

variable6 %= 5;

#endregion

// ПРАВИЛО:

// Для типов float и double, не происходит преобразования типа результата арифметических операций.

// Присвоение со сложением.

float variable7 = 0;

variable7 = variable7 + 5;

variable7 += 5;

// Присвоение с умножением.

double variable8 = 0;

variable8 = variable8 \* 5;

variable8 \*= 5;

// задержка

Console.ReadKey();

}

}

}

**Задачи для самостоятельного решения:**

**Задание 1**

Имеется 3 переменные типа int x = 10, y = 12, и z = 3;

Выполните и рассчитайте результат следующих операций для этих переменных:

* x += y - x++ \* z;
* z = --x – y \* 5;
* y /= x + 5 % z;
* z = x++ + y \* 5;
* x = y - x++ \* z;

Сформулируйте вывод о порядке действий

**Задание 2**

Используя Visual Studio, создайте проект по шаблону Console Application.

Вычислите среднее арифметическое трех целочисленных значений и выведите его на экран.

С какой проблемой вы столкнулись? Какой тип переменных лучше использовать для корректного

отображения результата?

**Задание 3**

Используя Visual Studio, создайте проект по шаблону Console Application.

Создайте константу с именем -pi (число π «пи»), создайте переменную радиус с именем – r. Используя формулу πR2, вычислите площадь круга и выведите результат на экран.

**Задание 4**

Используя Visual Studio, создайте проект по шаблону Console Application.

Напишите программу расчета объема - V и площади поверхности -S цилиндра.

Объем V цилиндра радиусом – R и высотой – h, вычисляется по формуле: V = πR2h

Площадь S поверхности цилиндра вычисляется по формуле: S = 2πR2 + 2πR(R+h)

Результаты расчетов выведите на экран.