

Типы данных:

Тип	Область значений	Размер
sbyte	-128 до 127	Знаковое 8-бит целое
byte	0 до 255	Беззнаковое 8-бит целое
char	U+0000 до U+ffff	16-битовый символ Unicode
bool	true или false	1 байт
short	-32768 до 32767	Знаковое 16-бит целое
ushort	0 до 65535	Беззнаковое 16-бит целое
int	-2147483648 до 2147483647	Знаковое 32-бит целое
uint	0 до 4294967295	Беззнаковое 32-бит целое
long	-9223372036854775808 до 9223372036854775807	Знаковое 32-бит целое
ulong	0 до 18446744073709551615	Беззнаковое 32-бит целое
float	$\pm 1,5 \cdot 10^{-45}$ до $\pm 3,4 \cdot 10^{33}$	4 байта, точность — 7 разрядов
double	$\pm 5 \cdot 10^{-324}$ до $\pm 1,7 \cdot 10^{306}$	8 байт, точность — 16 разрядов
decimal		12 байт, точность — 28 разрядов

Операции:

Операция	Назначение	Пример
+	Сложение	$X = x + y;$
-	Вычитание	$X = x - y;$
*	Умножение	$Z = x * y;$
/	Деление	$Z = x / y;$
%	Деление по модулю	$Z = t \% e;$

`Math.Pow(x, n)` возведение в степень n числа x (корень $1.0/n$)

например $\sqrt{8}$ `g = Math.Pow(8, 1.0/2);`

Деление по модулю возможно только для целых чисел. Оно обеспечивает получение остатка от деления двух целых чисел. Например, в результате операции `15 % 4` получится число 3. Таким образом, пара операций `/` и `%` обеспечивает для целых чисел специфическую возможность: можно получить и целую часть, и остаток от деления.

Арифметические операции можно объединять вместе с операцией присваивания, записывая две операции в виде одной. Это так называемая сокращённая форма записи.

Например:

обычная запись: $a = a + \text{Stoim};$

сокращённая форма: $a += \text{Stoim};$

К арифметическим операциям относятся две специфические операции – инкремент и декремент. Эти операции позволяют изменять значения переменных на 1. Инкремент – увеличивает, обозначается знаком ++. Декремент – уменьшает, обозначается знаком --. По сути дела, эти операции являются сокращённой формой записи для выражений типа $i = i + 1$ и $i = i - 1$. Например, вместо $a = a + 1$; можно записать: $a++$; (постфиксный вариант) или $++a$; (префиксный вариант). Аналогично: $--\text{Prim}$; $\text{Prim}--$;

Между двумя последними формами нет никакой разницы, если эти операции являются единственными в записи выражения, т.е. используются самостоятельно. Если же речь идёт об использовании в сложном выражении, то приходится учитывать, когда на самом деле выполняются эти операции. Суть в том, что при постфиксном варианте записи значение переменной сначала используется, а потом изменяется на 1. При префиксном, наоборот, сначала изменяется, а потом используется.

Например:

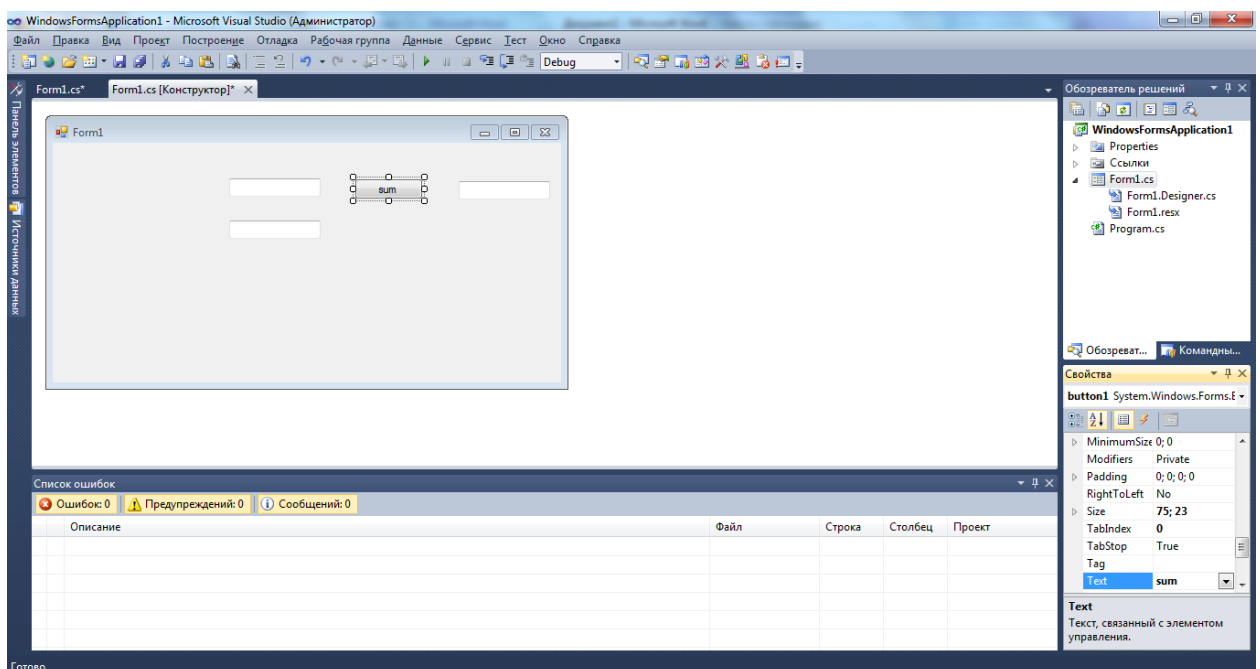
$t = 5; // t = 5$

$n = 4 * t++; // n = 20, t = 6$

$k = ++n / 3; // n = 21, k = 7$

$b = --k / t; // k = 6, b = 1$

Пример 1. Сложить два числа введённых в текстовые поля. Результат вывести в третье текстовое поле.



```
private void button1_Click(object sender, EventArgs e)
```

```
{  
    int c;  
    int a = Convert.ToInt32(textBox1.Text);  
    int b = Convert.ToInt32(textBox2.Text);  
    c=a+b;  
    textBox3.Text = Convert.ToString(c);  
}
```

Задание1: Найти сумму трех чисел, а так же произведение, разность, отношение, остаток от деления и возведение в степень двух чисел.