

ВВЕДЕНИЕ В C#

ВВЕДЕНИЕ

.NET Framework – это платформа, созданная Microsoft для разработки приложений.

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств.

ЯЗЫК ПРОГРАММИРОВАНИЯ C#

Язык программирования C# был разработан Microsoft на базе языков C и C++ специально для работы с платформой .NET.

На языке C# можно писать приложения:

- Windows-приложения (например, Microsoft Office);
- Web-приложения;
- Web-службы.

КОММЕНТАРИИ

В C# различают:

- многострочные комментарии

/*многострочный

комментарий*/

- однострочные комментарии

//однострочный комментарий

ИДЕНТИФИКАТОРЫ

Имена, или идентификаторы, служат для того чтобы обращаться к программным объектам и различать их.

В идентификаторе могут использоваться:

- буквы;
- цифры;
- символ подчеркивания.

Прописные и строчные буквы различаются.

Пример: *Hello, hello и hELlo* — три разных имени.

ИДЕНТИФИКАТОРЫ

Первым символом идентификатора может быть буква или знак подчеркивания, но не цифра.

Длина идентификатора не ограничена.

Пробелы внутри имен не допускаются.

Пример: `chislo1`, `chislo_1`

КЛЮЧЕВЫЕ СЛОВА

Ключевые слова — это зарезервированные идентификаторы, которые имеют специальное значение для компилятора. Их можно использовать только в том смысле, в котором они определены.

Пример: `if`, `else`, `try`, `void`

КОНСТАНТЫ

В C# константы (литераты) объявляются:

```
const <тип данных> <имя константы> = <значение константы>;
```

Пример: const int x = 55;

```
const double pi = 3.1415926535897932384626;
```

КОНСТАНТЫ

Константы в C# бывают:

- логические;
- целые;
- вещественные;
- символьные;
- строковые;
- константа null (ссылка, которая не указывает ни на какой объект).

ПЕРЕМЕННЫЕ

В C# переменные объявляются:

`<тип данных> <имя переменной>;`

Пример: `int x; float a;`

ТИПЫ ДАННЫХ

К простым типам относятся типы вроде числовых и булевских значений.

Целочисленные типы

Тип	Допустимые значения
sbyte	Целые числа от -128 до 127
byte	Целые числа от 0 до 255
short	Целые числа от -32768 до 32767
ushort	Целые числа от 0 до 65535
int	Целые числа от -2147483648 до 2147483647
uint	Целые числа от 0 до 4294967295
long	Целые числа от -9223372036854775808 до -9223372036854775807
ulong	Целые числа от 0 до 18446744073709551615

ТИПЫ ДАННЫХ

Типы переменных для хранения чисел с плавающей запятой (вещественные)

Тип	Приблиз. мин. значение	Приблиз. макс. значение
float	1.5×10^{-45}	3.4×10^{38}
double	5.0×10^{-324}	1.7×10^{308}
decimal	1.0×10^{-28}	7.9×10^{28}

Нечисловые простые типы

Тип	Допустимые значения
char	Одиночный символ Unicode, хранимый в виде целого числа от 0 до 65535
bool	Булево значение – true либо false
string	Последовательность символов

МОДИФИКАТОРЫ ДОСТУПА

- **public**: публичный, общедоступный класс или член класса. Такой объект доступен из любого места в коде, а также из других программ и сборок.
- **private**: закрытый класс или член класса. Представляет полную противоположность модификатору `public`. Такой закрытый объект доступен только из кода в том же классе или контексте.

Пример: `public int x; private float a;`

МАТЕМАТИЧЕСКИЕ ОПЕРАЦИИ

Операция	Категория	Пример	Результат
+	Бинарная	$v1=v2+v3$	$v1$ присваивается значение – результат сложения $v2$ и $v3$
-	Бинарная	$v1=v2-v3$	$v1$ присваивается значение – результат вычитания $v2$ и $v3$
*	Бинарная	$v1=v2*v3$	$v1$ присваивается значение – результат умножения $v2$ и $v3$
/	Бинарная	$v1=v2/v3$	$v1$ присваивается значение – результат деления $v2$ на $v3$
%	Бинарная	$v1=v2\%v3$	$v1$ присваивается значение – остаток от деления $v2$ на $v3$

ОПЕРАЦИИ ИНКРЕМЕНТА И ДЕКРЕМЕНТА

Операция	Категория	Пример	Результат
++	Унарная	$v1=++v2$	$v1$ присваивается значение – $v2+1$. $v2$ увеличивается на 1
--	Унарная	$v1=--v2$	$v1$ присваивается значение – $v2-1$. $v2$ уменьшается на 1
++	Унарная	$V1=v2++$	$v1$ присваивается значение – $v2$. $v2$ увеличивается на 1
--	Унарная	$v1=v2--$	$v1$ присваивается значение – $v2$. $v2$ уменьшается на 1

ПРЕОБРАЗОВАНИЕ ТИПОВ

- применяется, когда преобразование из типа А в тип В возможно при любых обстоятельствах, а правила выполнения преобразования достаточно просты для того, чтобы доверить их компилятору.
- Для преобразования типов используется класс `Convert`.

Пример: `Convert.ToBoolean(val)`

Команда	Результат
Convert.ToBoolean(val)	val, преобразованное в bool
Convert.ToByte(val)	val, преобразованное в byte
Convert.ToChar(val)	val, преобразованное в char
Convert.ToDecimal(val)	val, преобразованное в decimal
Convert.ToDouble(val)	val, преобразованное в double
Convert.ToInt16(val)	val, преобразованное в short
Convert.ToInt32(val)	val, преобразованное в int
Convert.ToInt64(val)	val, преобразованное в long
Convert.ToSByte(val)	val, преобразованное в sbyte
Convert.ToSingle(val)	val, преобразованное в float
Convert.ToString(val)	val, преобразованное в string
Convert.ToUInt16(val)	val, преобразованное в ushort
Convert.ToUInt32(val)	val, преобразованное в uint
Convert.ToUInt64(val)	val, преобразованное в ulong

СОБЫТИЯ, СВОЙСТВА, МЕТОДЫ

События сигнализируют системе о том, что произошло определенное действие. И если нам надо отследить эти действия, то как раз мы можем применять события.

Для событий генерируются обработчики событий, в которых пишется код, выполняющийся в случае их возникновения.

СОБЫТИЯ, СВОЙСТВА, МЕТОДЫ

Методами - функция как подпрограмма..

```
[модификаторы] тип_возвращаемого_значения ИмяМетода([параметры])  
{  
    // Тело метода  
}
```

Пример:

```
public int MyMethod(int x)  
{  
    int s = 2;  
    int summa = s + x;  
    return summa;  
}
```

СОБЫТИЯ, СВОЙСТВА, МЕТОДЫ

Свойство - сочетает в себе поле с методами доступа к нему.

Пример: `textBox1.Text`

Вызвать метод либо свойство у объекта (элемента на форме) можно написав его имя (свойство Name), а затем поставить ТОЧКУ. В итоге откроется меню с доступными свойствами и методами для данного объекта.

