

В.В. Подбельский

Иллюстрации к курсу лекций по дисциплине «Программирование»

01

Программа на C#. Элементы синтаксиса

«Каркас» простой программы

```
// «Каркас» простой программы
class Program { // Класс приложения
    static void Main() { // Точка входа
        // Код решения задачи
    } // Main()
} // class Program
```

Детализация программы (шаг 1)

**// Задание: Ввести вещественное число, вывести
// целую и дробную части числа.**

```
class Program { // Класс приложения
    static void Main() { // Точка входа
        double real; // Вещественное число
        string input; // Вводимая строка
// Ввести число, обработать, вывести результат
        System.Console.WriteLine("Выход - ENTER");
        System.Console.ReadLine();
    } // Main()
} // class Program
```

Ввести число, обработать, вывести результат

```
Console.Write("Введите число:");
input = Console.ReadLine();
if (!double.TryParse(input, out real))
    Console.WriteLine("Ошибка ввода");
else
{
    int integer = (int)real;    // Целая часть
    double fraction = real - integer;    // Дробная часть
    System.Console.WriteLine(
        "integer = {0}; fraction = {1}",
        integer, fraction);
    Console.WriteLine("целая часть: " + ((int)real));
    Console.WriteLine("дробная часть: " + (real - (int)real));
}
```

Результат выполнения:

Введите вещественное число: 4,9

integer = 4; fraction = 0,9

Для выхода нажмите ENTER.

Структура простейшей программы на C#

using-директивы

class имя_класса {

Поля класса . . .

Методы класса . . .

static void Main () {

Операторы метода

. . . .

} // End of Main

} // End of class

Роль декларации using

```
System.Console.WriteLine("Для выхода ENTER.");  
System.Console.ReadLine();
```

```
using System;
```

```
. . . . .
```

```
Console.WriteLine("Для выхода ENTER.");  
Console.ReadLine();
```

Идентификаторы C#

Первый символ : буква, подчеркивание, @

Не первый символ: буква, цифра, подчеркивание

Правильные:

@if

_007

якорь

s_2_4_6

день_недели

Ошибочные:

if

666

номер дома

name@

mail.ru

Служебные слова C#

abstract	as	base	bool	break	byte
case	catch	char	checked	class	const
continue	decimal	default	delegate	do	double
else	enum	event	explicit	extern	false
finally	fixed	float	for	foreach	goto
if	implicit	in	int	interface	internal
is	lock	long	namespace	new	null
object	operator	out	override	params	private
protected	public	readonly	ref	return	sbyte
sealed	short	sizeof	stackalloc	static	string
struct	switch	this	throw	true	try
typeof	uint	ulong	unchecked	unsafe	ushort
using	virtual	void	volatile	while	

Контекстно-зависимые служебные слова

**add
equals
into
partial
var**

**ascending
from
join
remove
where**

**by
get
let
select
yield**

**descending
global
on
set**

**dynamic
group
orderby
value**

Пробельные символы

- Пробел
- Табуляция
- Символ начала новой строки
- Символ возврата каретки

Метод Main()

```
static void Main( )  
{  
    Операторы  
}
```

```
static int Main( )  
{  
    Операторы  
}
```

Блок и оператор в С#

```
{  
    int группа = 5;  
    Console.WriteLine("Номер группы: "+группа);  
}
```

Комментарии в коде на C#

- Однострочные:

// Это однострочный комментарий

- С двумя ограничителями:

```
int b=0, /* начало */ e=1; /* конец */
```

- Документирующие:

```
/// <тег>
```

```
/// Этот текст будет содержимым
```

```
/// элемента XML-документа
```

```
/// </тег>
```

Консольный ввод-вывод данных

Средства вывода:

```
System.Console.Write("text1+");  
System.Console.WriteLine("text2");
```

Чтение строки:

```
string line = System.Console.ReadLine();
```

Получение вещественного значения:

```
double.TryParse(строковое_представление, out переменная);
```

Получение целого значения:

```
int.TryParse(строковое_представление, out переменная);
```