Лекция 7

Алфавит и лексика языка программирования. Структура программы

Цель: 1 Познакомиться с основными элементами языка C#

2 Научиться составлять программы на языке C#

Таблица 1

Прописные буквы латинского алфавита	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Строчные буквы латинского алфавита	a b c d e f g h i j k l m n o p q r s t u v w x y z
Символ подчеркивания	_
Символ	@

Таблица 2

Прописные буквы русского алфавита	А Б В Г Д Е Ж З И К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ы Ь Э Ю Я
Строчные буквы русского алфавита	а б в г д е ж з и к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я
Арабские цифры	0 1 2 3 4 5 6 7 8 9

Таблица 3

Символ	Наименование	Символ	Наименование
,	запятая	)	круглая скобка правая
.	точка	(	круглая скобка левая
;	точка с запятой	}	фигурная скобка правая
:	двоеточие	{	фигурная скобка левая
?	вопросительный знак	<	меньше
'	апостроф	>	больше
!	восклицательный знак	[	квадратная скобка
|	вертикальная черта	]	квадратная скобка
/	дробная черта	#	номер
\	обратная черта	%	процент
~	тильда	&	амперсант
*	звездочка	^	логическое не
+	плюс	=	равно
-	минус	"	кавычки

Управляющие и разделительные символы

- пробел;
- символ табуляции;
- символ перевода строки;
- символ возврата каретки;
- новая страница;
- новая строка.

Управляющие последовательности

Управляющая последовательность	Наименование	Шестнадцатеричная замена
\a	Звонок	007
\b	Возврат на одну позицию	008
\t	Горизонтальная табуляция	009
\n	Переход на новую строку	00A
\v	Вертикальная табуляция	00B
\r	Возврат каретки	00C
\f	Перевод формата (для перехода к началу следующей странице)	00D

Управляющая последовательность	Наименование	Шестнадцатеричная замена
\"	Кавычки	022
\'	Апостроф	027
\0	Ноль-символ	000
\\	Обратная дробная черта	05C
\ddd	Символ набора кодов ПЭВМ в восьмеричном представлении	
\xddd	Символ набора кодов ПЭВМ в шестнадцатеричном представлении	

Примеры

- \r - общая управляющая последовательность,
- \015 - восьмеричная управляющая последовательность,
- \x00D - шестнадцатеричная управляющая последовательность,
- "ABCDE\x009FGH" - два слова ABCDE FGH, разделенные
- 8-ю пробелами,
- "ABCDE\x09FGH" - на печати появится ABCDE=|=GH, так как компилятор воспримет последовательность \x09F как символ "=|=",
- \h - представляется символом h в строковой или символьной константе,
- "За \

  морем" - строковый литерал "За морем"

Идентификаторы - это имена констант, переменных, меток, типов, объектов, классов, свойств, функций, модулей, программ, полей в структурах и т. д..

Правила формирования идентификатора

1. Для образования идентификаторов могут быть использованы строчные или прописные буквы латинского алфавита, арабские цифры, символ «подчеркивание» (_) и символ @.
2. Нельзя использовать цифры в качестве первого символа идентификатора.
3. Не следует использовать символ «подчеркивание» (_) в качестве первого символа идентификатора.
4. Символ @ можно использовать только в качестве первого символа идентификатора.
5. Идентификатор не должен совпадать с ключевыми словами, с зарезервированными словами и именами функций библиотеки компилятора языка C#.

Правила формирования идентификатора

6. Длина идентификатора не ограничена. Пробелы внутри имен не допускаются.
7. Язык регистрозависимый.
8. В идентификаторах C# разрешается использовать, помимо латинских букв, буквы национальных алфавитов. Например, правильными являются идентификаторы Фёкла и calc. Более того, можно применять даже так называемые *escape-последовательности Unicode*, то есть представлять символ с помощью его кода в шестнадцатеричном виде с префиксом \u, например, \u00F2.

Ключевые слова - это зарезервированные идентификаторы, которые наделены определенным смыслом.

Ключевые 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	

Ключевые слова языка C++

asm	auto	bool	break
case	catch	char	class
const	const_cast	continue	default
delete	do	double	dynamic_cast
else	enum	explicit	export
extern	false	float	for
friend	goto	if	inline
int	long	mutable	namespace
new	operator	private	protected
public	register	reinterpret_cast	return
short	signed	sizeof	static
static_cast	struct	switch	template
this	throw	true	try
typedef	typeid	typename	union

Использование комментариев в тексте программы

Комментарий - это набор символов, размещенный между парами символов /* и */, который игнорируется компилятором. В языке C# добавляется возможность однострочного комментария с помощью символов //.

Например:

```
/* комментарии к программе */
```

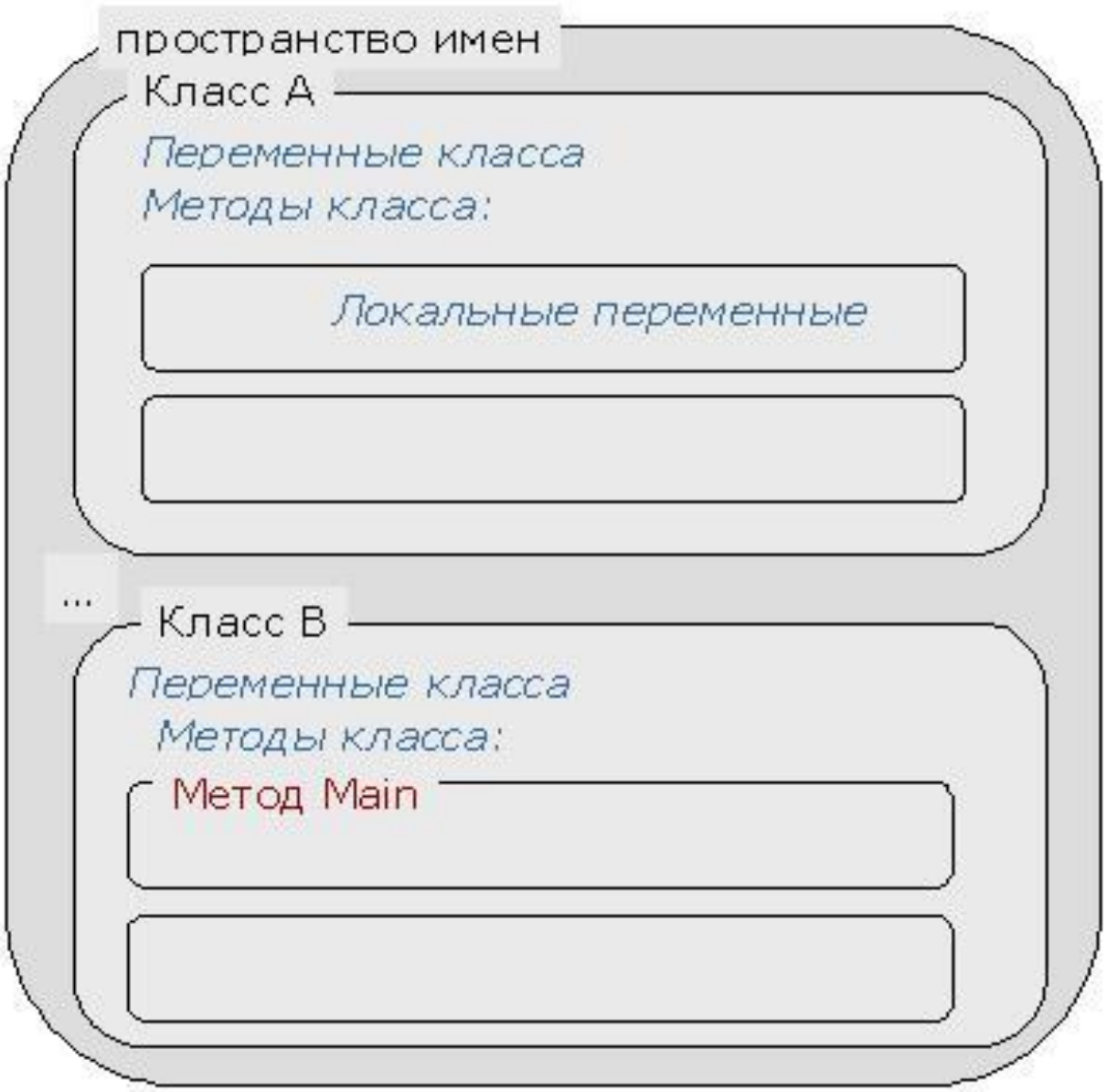
или

```
/* надо быть осторожным, чтобы внутри последовательности, которая игнорируется компилятором, не попались операторы программы, которые также будут игнорироваться */
// все символы до конца строки считаются комментарием
```

Неправильное определение комментариев

```
/* комментарии к алгоритму /* решение краевой задачи */ */
/* комментарии к алгоритму решения */ краевой задачи */
```

Структура программы



Пример 1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            int @if,  Ёж;
            Ёж = 10;
            Console.Write("Введите число -  ");
            @if = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("!!Привет!!");
            Console.WriteLine(" @if = {0},    Ёж = {1}", @if, Ёж); // вариант 1
            Console.WriteLine("@if = " + @if + ",  Ёж = " + Ёж); // вариант 2
        }
    }
}
```

Пример 2

```
using System; // …

namespace ConsoleApp1
{
   class Program
{
       public static string Hello()    // метод - функция
       {
           return "Hell to World";
       }

      static void Main(string[] args)
      {
          string message = Hello();        // вызов метода
          Console.WriteLine(message);
          Console.WriteLine("!!!!!!!");
          Console.ReadKey();  // остановка экрана
       }
   }
}
```

Контрольные вопросы

1 Какие управляющие и разделительные символы входят в состав языка C#?

2 Какие управляющие последовательности входят в состав языка C#?

3 Что такое идентификатор, и по каким правилам он формируется?

4 Какие структурные элементы являются обязательными блоками программы?

Домашнее задание

1 Напишите пять правильных и пять ошибочных определений идентификаторов. С помощью комментариев оформите объяснение причин ошибок в определениях идентификаторов.

2 Напишите программу на языке C++, выводящую на экран сумму двух целых чисел, введенных пользователем.