

Лекция 3.

Основы языка C#.

Алфавит C#

2

- Алфавит C# включает:
 - строчные и прописные буквы латинского алфавита;
 - цифры от 0 до 9;
 - символ “_” (подчеркивание);
 - набор специальных символов: " { } , [] + - % / \ ; ' : ? < > = ! & # * ;
 - комментарии: /* */ (многострочные) и // (однострочные).
- Алфавит C# служит для построения слов – лексем:
 - идентификаторы;
 - ключевые слова;
 - знаки (символы) операций;
 - литералы;
 - разделители (пробелы, знаки табуляции и переход на новую строку).

Лексемы.

Идентификаторы и ключевые

3

слова

- Правила построения идентификаторов:
 - первый символ – буква;
 - второй и далее символы – буквы, цифры, «_»;
 - длина не ограничена.
- Ключевые слова:

bool	class	else	goto	protected	sizeof	true
break	const	false	if	public	string	try
case	continue	finally	int	readonly	struct	typeof
catch	default	float	new	ref	switch	void
char	do	for	null	return	this	while
checked	double	foreach	private	short	throw	

Лексемы.

Литералы

4

- В C# существует четыре типа литералов:
 - целочисленный литерал:
 - десятичные, восьмеричные (0 - префикс), шестнадцатеричные (0X или 0x - префикс);
 - беззнаковые (U или u), длинные (L или l) - суффикс.
 - вещественный литерал:
 - вещественный (F/f), длинный вещественный (L/l), с мантиссой (E/e).
 - символьный литерал: 'f', 'H', '8', '+', ':';
 - строковый литерал: "string", "мама мыла раму".

Встроенные типы данных

5

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	-32 768 до 32 767	Знаковое 16-бит целое
ushort	0 до 65 535	Беззнаковое 16-бит целое
int	-2 147 483 648 до 2 147 483 647	Знаковое 32-бит целое
uint	0 до 4 294 967 295	Беззнаковое 32-бит целое
long	-9,223,372,036,854,775,808 до 9,223,372,036,854,775,807 $\pm 1.5 \cdot 10^{43}$ до $\pm 3.4 \cdot 10^{38}$	Знаковое 32-бит целое
float	$\pm 5.0 \cdot 10^{-324}$ до $\pm 1.7 \cdot 10^{308}$	4 байта, точность — 7 разрядов
double		8 байт, точность — 16 разрядов
decimal		12 байт, точность — 28 разрядов

Преобразование встроенных ТИПОВ ДАННЫХ

6

- Неявное преобразование – автоматическое:

```
short x = 1; int y = x; // неявное преобразование
```

```
short x; int y = 5; x = y; // не скомпилируется
```

- Явное преобразование:

```
short x; int y = 5; x = (short) y; // скомпилируется
```

Переменные

7

- Переменная – именованная область памяти определенного типа, значение которой может изменяться программно.

тип имя_перем; // объявление переменной

имя_перем = нач_знач; // инициализация, если сразу после объявления; присвоение значения - при дальнейшем использовании

тип имя_перем = нач_знач; // определение переменной

```
int x, y, c;
```

```
x = 5;
```

```
int z = 8, p = 456;
```

Константы

8

- Константа – переменная, значение которой не может быть изменено.

```
const ТИП ИМЯ_КОНСТ=нач_знач; // определение  
    КОНСТАНТЫ
```

```
const double pi = 3.141592;
```

```
y = x * pi;
```

```
z = 5 * y * pi;
```


Выражения

- Выражения состоят из операндов – констант, переменных, функций, – объединенных знаками операций и скобками.
- Примеры выражений: $a+b$ $12.5 - z$ $2*(X+Y)$

$x++$

$x+++b$

$--n*2$

$n*=1$

Приоритет	Операции	Приоритет	Операции
1. Унарные	+ - ! ~ ++x --x (T)x	8. Логическое исключающее ИЛИ	^
2. Мультиплика- тивные	* / %	9. Логическое ИЛИ	
3. Аддитивные	+ -	10. Условное И	&&
4. Сдвиг	<< >>	11. Условное ИЛИ	
5. Отношения, проверка типов	< > <= >= is as	12. Условное выражение	? :
6. Эквивалентность	== !=	13. Присваивание	= *= /= %= += -= <<= >>= &= ^= =
7. Логическое И	&		

Операции

10

- Арифметические: **+** (бинарная, унарная), **-** (бинарная, унарная), *****, **/**, **%** (остаток от деления), **++** (инкремент, унарная), **--** (декремент, унарная)

$1/5 \rightarrow 0$ $5./3$ или $5./3.$ или $5/3 \rightarrow 1.666666$

`x=x+1;` `++x;` `x++;` // одинаковый результат

- Оп. явного приведения типов:

`int x = 1, y = 5; float z = x/y; // z = 0`

`int x = 1; float y = 5; float z = x/y; // z = 0.2`

`int x = 1, y = 5; float z = (float)x/y; // z = 0.2`

- Оп. отношения: **==** (равно), **!=** (не равно), **<**, **>**, **<=**, **>=**

- Логические: **!** (не), **&&** (и), **| |** (или)

Операции (продолжение)

11

□ Оп. присваивания: =

$a = b = c = x + y;$

$a += 2$ эквивалентно $a = a + 2$

$x -= a + b$ эквивалентно $x = x - (a + b)$

$p /= 10$ эквивалентно $p = p / 10$

`int a = 3; int b = 2; int c = ++a * b++; // a = 4, b = 3, c = 8`

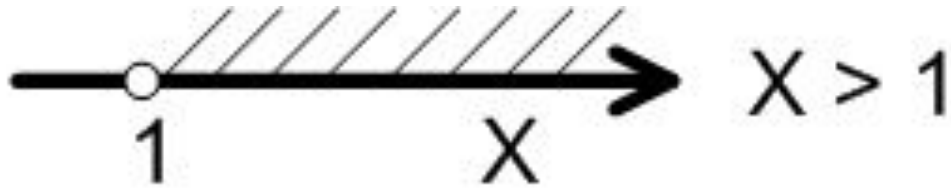
□ Оп. «условие» (3 операнда): $\text{выр1} \text{ ? } \text{выр2} \text{ : } \text{выр3}$

$X < 0 \text{ ? } -X \text{ : } X;$ // модуль числа

$\text{max} = (a \leq b) \text{ ? } b \text{ : } a;$ // максимальное значение

Пример 1

12



Пример 2

13

№	Задача	Ответ
1	<pre>int a = 5, b = 3; int c = --a + b;</pre>	a= b= c=
2	<pre>int a = 5, b = 3; int c = ++a + b--;</pre>	a= b= c=
3	<pre>int a = 5, b = 3; int c = (a++ + b++) * 2;</pre>	a= b= c=
4	<pre>int a = 5, b = 3; int c = a > b ? a : b;</pre>	a= b= c=

Пример 2 (ответы)

14

№	Задача	Ответ
1	<pre>int a = 5, b = 3; int c = --a + b;</pre>	a=4 b=3 c=7
2	<pre>int a = 5, b = 3; int c = ++a + b--;</pre>	a=6 b=2 c=9
3	<pre>int a = 5, b = 3; int c = (a++ + b++) * 2;</pre>	a=6 b=4 c=16
4	<pre>int a = 5, b = 3; int c = a > b ? a : b;</pre>	a=5 b=3 c=5

Функции (методы)

15

- Функции (методы) задают определенную функциональность, которую можно многократно выполнять при различных значениях параметров
- Функции (методы) состоят из заголовка и тела функции

**[модификаторы] тип_результата_функции имя_функции
([список_формальных_аргументов])**

{

// операторы

}

private int sum(int a, int b) // заголовок функции

{ return a + b; } // тело функции

16

Спасибо за внимание

Вопросы...