

Богатов Р.Н.

Программирование на языке высокого уровня

Лекция 9.

Символы. Кодировки. Строки. Массивы строк

Кафедра АСОИУ ОмГТУ, 2012

Встроенные типы данных C#

Тип	Диапазон значений	Размер (байт)
sbyte	-128 ... 127	1
byte	0 ... 255	1
short	-32 768 ... 32 767	2
ushort	0 ... 65 535	2
int	-2 147 483 648 ... 2 147 483 647	4
uint	0 ... 4 294 967 295	4
long	-9 223 372 036 854 775 808 ... 9 223 372 036 854 775 807	8
ulong	0 ... 18 446 744 073 709 551 615	8
float	$\pm 1,5 \cdot 10^{-45} \dots \pm 3,4 \cdot 10^{+38}$, точность – 7 разрядов	4
double	$\pm 5,0 \cdot 10^{-324} \dots \pm 1,7 \cdot 10^{+308}$, точность – 16 разрядов	8
decimal	$\pm 1 \cdot 10^{-28} \dots \pm 7,9 \cdot 10^{+28}$, точность – 28 разрядов	16
bool	true или false	1
char	0 ... 65 535 (код символа)	2

СИМВОЛЫ

- Сколько-значный должен быть код для представления одного символа текста?
- **A...Z a...z 0...9 " ! ? . , ; № _ - + * / % \ = < > () { } [] @ # \$ ^ & ~ | '**
– это $27+27+10+32+\dots \geq 96$ символов
- 6 бит = 64 значения
7 бит = 128 значений
8 бит = 256 значений
- Изначальная идея: **один символ – один байт**. Коды 0...127 – основной набор символов, коды 128...255 – для национального алфавита.
- ASCII – базовая 7-битовая кодировка для латинского алфавита и **ОСНОВНЫХ СИМВОЛОВ**. Кириллические кодировки (CP866, Windows CP1251, KOI8-R и другие) отличаются только размещением русских букв в диапазоне кодов 128...255.

Кодировка Windows CP1251

0		16		32		48	0	64	@	80	P	96	`	112	p
1		17		33	!	49	1	65	A	81	Q	97	a	113	q
2		18		34	"	50	2	66	B	82	R	98	b	114	r
3		19		35	#	51	3	67	C	83	S	99	c	115	s
4		20		36	\$	52	4	68	D	84	T	100	d	116	t
5		21		37	%	53	5	69	E	85	U	101	e	117	u
6		22		38	&	54	6	70	F	86	V	102	f	118	v
7		23		39	'	55	7	71	G	87	W	103	g	119	w
8		24		40	(56	8	72	H	88	X	104	h	120	x
9		25		41)	57	9	73	I	89	Y	105	i	121	y
10		26		42	*	58	:	74	J	90	Z	106	j	122	z
11		27		43	+	59	;	75	K	91	[107	k	123	{
12		28		44	,	60	<	76	L	92	\	108	l	124	
13		29		45	-	61	=	77	M	93]	109	m	125	}
14		30		46	.	62	>	78	N	94	^	110	n	126	~
15		31		47	/	63	?	79	O	95	_	111	o	127	□

128	Ђ	144	ђ	160		176	°	192	A	208	P	224	a	240	p
129	Ѓ	145	ѓ	161	Ў	177	±	193	Б	209	С	225	б	241	с
130	Д	146	д	162	ў	178	ł	194	В	210	Т	226	в	242	т
131	Ѓ	147	ѓ	163	Ј	179	і	195	Г	211	У	227	г	243	у
132	„	148	”	164	ѣ	180	ŕ	196	Д	212	Ф	228	д	244	ф
133	…	149	•	165	Ѓ	181	μ	197	Е	213	Х	229	е	245	х
134	†	150	—	166	‡	182	¶	198	Ж	214	Ц	230	ж	246	ц
135	‡	151	—	167	§	183	·	199	З	215	Ч	231	з	247	ч
136	€	152	□	168	Ё	184	ё	200	И	216	Ш	232	и	248	ш
137	‰	153	™	169	©	185	№	201	Й	217	Щ	233	й	249	щ
138	Љ	154	љ	170	Є	186	є	202	К	218	Ъ	234	к	250	ъ
139	‹	155	›	171	«	187	»	203	Л	219	Ы	235	л	251	ы
140	Њ	156	њ	172	–	188	ј	204	М	220	Ь	236	м	252	ь
141	Ќ	157	ќ	173	–	189	ѕ	205	Н	221	Э	237	н	253	э
142	Ћ	158	ћ	174	®	190	ѕ	206	О	222	Ю	238	о	254	ю
143	Ќ	159	ќ	175	İ	191	ı	207	П	223	Я	239	п	255	я

Кодировка Unicode

Символ

Символы Специальные знаки

Шрифт: (обычный текст)

Набор: глаголица

128...65535 (16 бит)

Ранее использовавшиеся символы:

Г · ± ∈ ∇ π × ≠ ∞ → € £ ¥ © ® ™ ≤ ≥ ÷ μ α β Ω Σ ☺ ☹ § † ‡ ... % é ÷ ñ ←

Код знака: UCS2 из: Юникод (шестн.)

Автозамена... Сочетание клавиш... Сочетание клавиш: UCS2, Alt+X

Вставить Отмена

Вставить Закреть

Символы в C#. Примеры

```
// объявление и инициализация  
char a, b;
```

```
a = '7';  
textBox1.Text += a;
```

```
b = (char)55;  
textBox1.Text += b;
```

```
// результат одинаков  
for (int i = 1; i < 256; i++)  
    textBox1.Text += (char)i;
```

```
// посмотреть кодировку ASCII
```

```
// посмотреть кодировку Unicode
```

```
for (int i = 1; i < 65536; i++)  
    textBox1.Text += (char)i;
```

```
// не запускайте -
```

```
// работает слишком долго :o(
```

Строки в C#. Примеры

```
// объявление и инициализация
```

```
string s1, s2, s3;  
s1 = "";  
s2 = "7"; // не эквивалентно  
s1 += "Привет, ";  
s3 = s1 + s2 + "!\r\nПока!";  
  
textBox1.Text = s3;
```

```
// сравнение строк
```

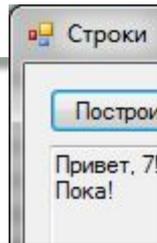
```
if (s1 == s2)  
    MessageBox.Show("Ура!");  
  
bool SuperUser;  
if (s3 == "Admin")  
    SuperUser = true;  
else  
    SuperUser = false;
```

```
// длина строки, посимвольный  
просмотр, переворот
```

```
for (int i = 0; i < s1.Length;  
i++)  
    s2 = s1[i] + s2;
```

```
// посимвольный  
просмотр, переворот
```

```
foreach (char x in s1)  
    s2 = x + s2;
```



Строки в C#. Подстроки

```
// поиск подстроки  
string s = textBox1.Text;
```

```
int p = s.IndexOf("Вася");  
if (p == -1)  
    MessageBox.Show("Вася не найдена.");  
else  
    MessageBox.Show("Вася найдена.");
```

```
// выделение подстроки
```

```
string s1 = "Дано: x = -3,1415; y = 0,5. Найти z." ;
```

```
int p1 = s1.IndexOf("Дано:");  
int p2 = s1.IndexOf("Найти z.");
```

```
string s1 = textBox1.Text;  
string s2 = textBox2.Text;  
double x = 0;
```

```
int p = s1.IndexOf(s2);  
if (p != -1)  
    textBox3.Text = s1.Insert(p, "НаШёЛ->");
```

Строки

Начальная строка: Привет! Как твои дела, Вася? Пока...

Искомая строка: Вася

Обработать

Результат: Привет! Как твои дела, НаШёЛ->Вася? Пока...

Кодировка Unicode



Константный массив строк

Задача: заменить в тексте каждую цифру на её название, взятое в скобки. Например, текст «Мне 18 лет» должен преобразоваться в «Мне (один)(восемь) лет».

```
string s = textBox1.Text;
string z = "";

foreach(char c in s)
    switch (c)
    {
        case '1':
            z += "(один) ";
            break;
        case '2':
            z += "(два) ";
            break;
        ...
        case '0':
            z += "(ноль) ";
            break;
        default:
            z += c;
            break;
    }

textBox2.Text = z;
```

```
string s = textBox1.Text;
string z = "";
string[] x = new string[] { "ноль", "один",
    "два", "три", "четыре", "пять", "шесть",
    "семь", "восемь", "девять" };

foreach(char c in s)
    if (c >= '0' && c <= '9')
        z += "(" + x[c - '0'] + ") ";
    else
        z += c;

textBox2.Text = z;
```

Замена цифр

Исходный текст
Агент 007 родился в

Изменить

Результат:
Агент (ноль)(ноль)(семь) родился в (один)(девять)(три)(семь) году.

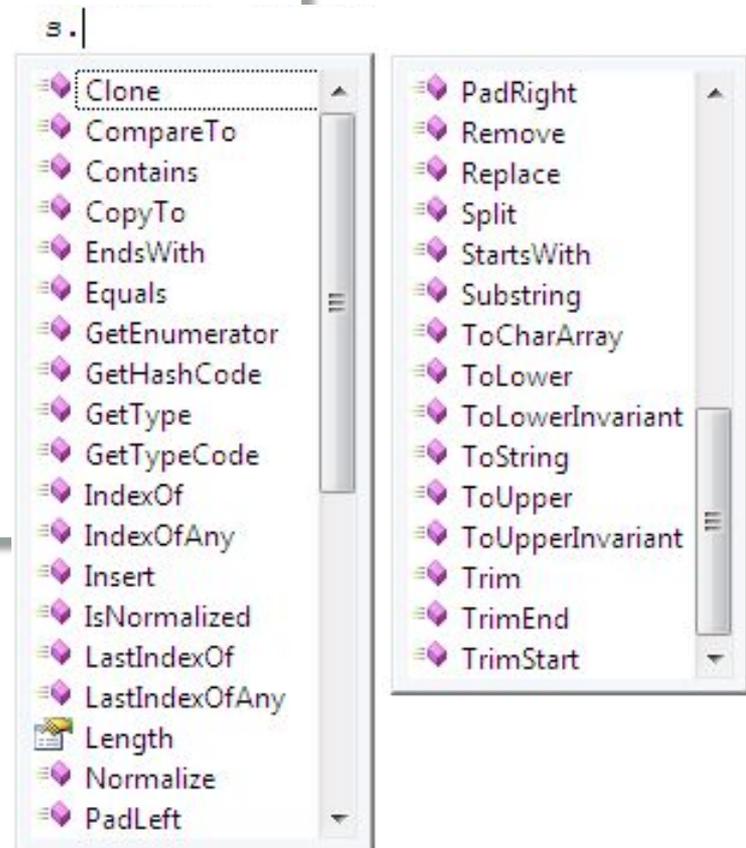
Строки в C#. Удаление, замена подстроки и прочее

```
string s = textBox1.Text;
```

```
// удаление  
string s = textBox1.Text;  
s = s.Re
```

```
s = s.Replace("0", "(ноль)");  
// замена  
s = s.Replace("1", "(один)");  
s = s.Replace("2", "(два)");  
s = s.Replace("3", "(три)");  
// замена  
s = s.Replace("4", "(четыре)");  
s = s.Replace("5", "(пять)");  
s = s.Replace("6", "(шесть)");  
s = s.Replace("7", "(семь)");  
s = s.Replace("8", "(восемь)");  
s = s.Replace("9", "(девять)");
```

```
textBox2.Text = s;
```



Строки в C#. Метод Split

```
// разбить текст в тех местах, где стоят знаки .?!  
string s = textBox1.Text;
```

```
// массив символов-разделителей вводит пользователь
```

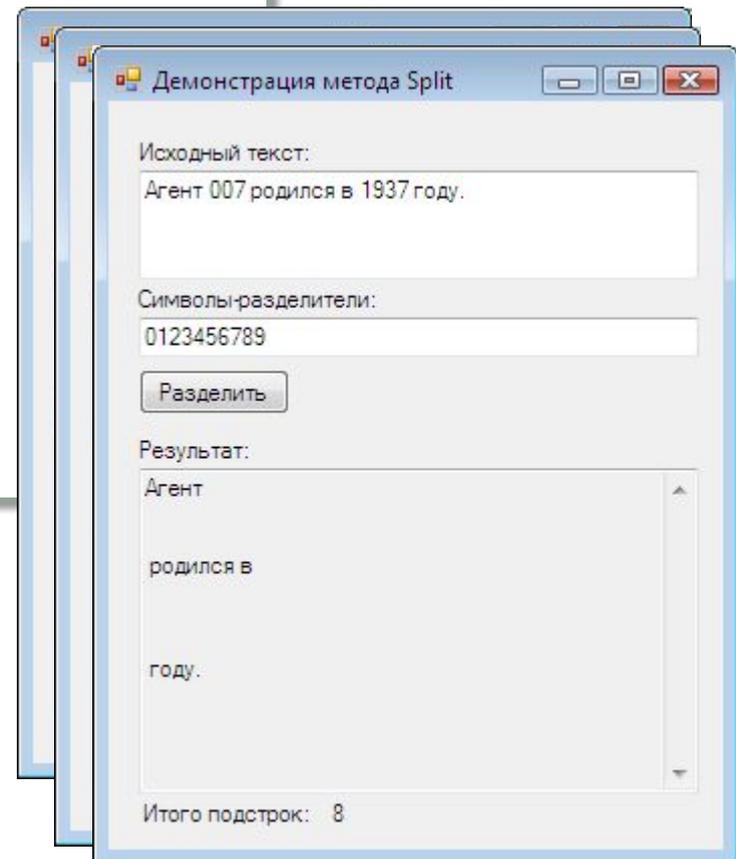
```
string s = textBox1.Text;  
char[] x = textBox2.Text.ToCharArray();
```

```
string[] r = s.Split(x);
```

```
// полученные строки выводим в столбик
```

```
textBox3.Text = "";  
foreach (string a in r)  
    textBox3.Text += a + "\r\n";
```

```
label4.Text = Convert.ToString(r.Length);
```



Пример: подсчёт гласных букв в тексте

```
string s = textBox1.Text.ToLower();  
string x = "aeёиоуыэя";  
  
int count = 0; // кол-во гласных в тексте  
foreach(char c in s)
```

```
string s = textBox1.Text.ToLower();  
char[] x = { 'a', 'e', 'ё', 'и', 'o', 'y', 'ы', 'э', 'ю', 'я' };  
  
int count = 0;  
for(int pos = -1; ; count++)  
{  
    pos = s.IndexOfAny(x, pos + 1);  
    if (pos == -1) break;  
}
```

labe

```
string s = textBox1.Text.ToLower();  
char[] x = { 'a', 'e', 'ё', 'и', 'o', 'y', 'ы', 'э', 'ю', 'я' };  
  
string[] r = s.Split(x);  
  
label3.Text = Convert.ToString(r.Length - 1);
```

label3

Домашнее задание

Решить задачу с помощью массива строк. Определить, в каком квадранте или на какой оси координат находится точка, заданная пользователем.

Если ответ имеет вид «Точка $(x_0; y_0)$ находится ...», то можно использовать массив строк с вариантами ответов: «на оси OY», «на оси OX», «в III квадранте», «в II квадранте», «в IV квадранте», «в I квадранте».