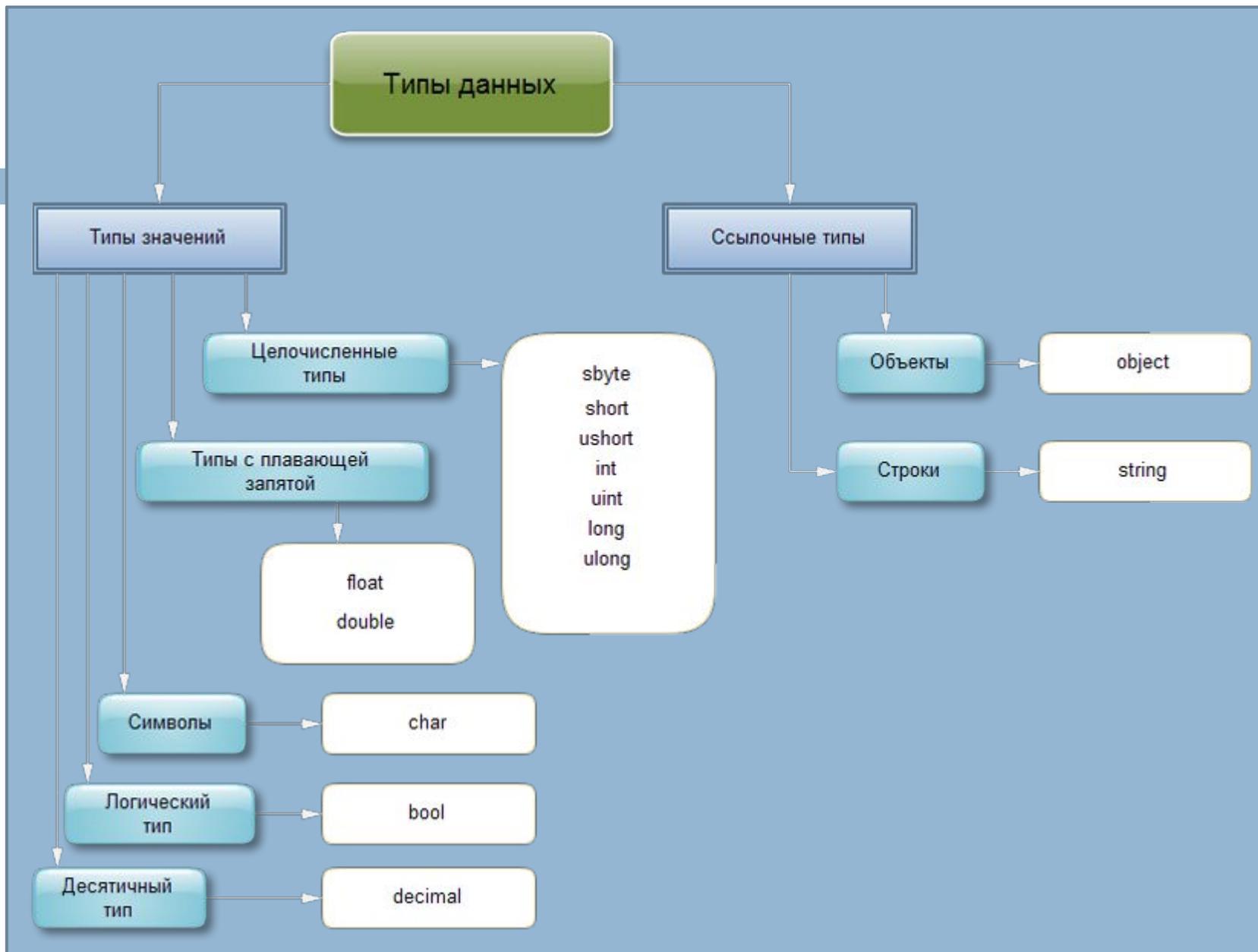


Лекция 18.

Строки



СИМВОЛЫ (char)

3

- 16-разрядные значения без знака в пределах от 0 до 65 535 (*уникод (Unicode)*)

```
char ch;
```

```
ch = 'z';
```

- **Управляющие последовательности СИМВОЛОВ**

Управляющая последовательность	Описание
\a	Звуковой сигнал (звонок)
\b	Возврат на одну позицию
\n	Новая строка (перевод строки)
\t	Горизонтальная табуляция
\'	Одинарная кавычка
\\	Обратная косая черта

Коды символов Юникод (таблица)

<http://ir2.ru/charcod5.htm>

4

34 "	153 ™	282 Ě 283 ě	407 Ĩ 408 Ķ	608 ğ	701 ˘	794 ˘ 795 ˘	895 ☞	1039 Ц
35 #	154 š 155 ›	284 Ğ 285 ğ	409 ĩ 410 ï	609 ğ	702 ˙	796 ˘ 797 ˘	896 ☞	1040 А
36 \$	156 œ 157 Ÿ	286 Ğ 287 ğ	411 ĺ 412 ш	610 ğ	703 ˚	798 ˘ 799 ˘	897 ☞	1041 Б
37 %	158 ž 159 Ÿ	288 Ğ 289 ğ	413 Ń 414 ŋ	611 ŷ	704 ˛	800_801_	898 ☞	1042 В
38 &	160 ı 161 ĵ	290 Ğ 291 ğ	415 Θ	612 ŷ	705 ˜	802_803_	899 ☞	1043 Г
39 '	162 ¢ 163 £	292 Ĥ 293 ĥ	416 Œ 417 œ	613 ŷ	706 <	804_805_	900 ˘	1044 Д
40 (164 ¤ 165 ¥	294 Ħ 295 ħ	418 Œ	614 ħ	707 >	806_807_	901 ˘	1045 Е
41)	166 167 §	296 Ĩ 297 ĩ	419 Œ	615 ħ 616 ħ	708 ^	808_809_	902 А	1046 Ж
42 *	168 ¨ 169 ©	298 Ĩ 299 ĩ	420 P 421 ß	617 ı 618 ı	709 v	810_811_	903 ˘	1047 З
43 +	170 ª 171 «	300 Ĩ 301 ĩ	422 R 423 Ž	619 ĩ 620 ĩ	710 ^	812_813_	904 Ё	1048 И
44 ,	172 – 173	302 Ĩ 303 ĩ	424 ¢ 425 Σ	621 ĩ 622 ĩ	711 ˇ	814_815_	905 H	1049 Й
45 -	174 ® 175 ¯	304 Ĩ 305 ĩ	426 ĵ 427 ĵ	623 w	712 ˘	816_817_	906 I	1050 К
46 .	176 ° 177 ±	306 Ĩ 307 ĩ	428 T 429 t	624 w	713 ˘	818_819_	907 ☞	1051 Л
47 /	178 ² 179 ³	308 Ĩ 309 ĩ	430 T	625 w	714 ˘	820~821~	908 O	1052 М
48 0	180 ´ 181 µ	310 K 311 k	431 U	626 w	715 ˘	822~823~	909 ☞	1053 Н
49 1	182 ¶ 183 ·	312 K 313 L	432 U 433 U	627 ŋ	716 ˘	824/825_	910 Y	1054 О
50 2	184 ˘ 185 ˘	314 Ĩ 315 ĩ	434 U	628 N	717 ˘	826_827_	911 Ω	1055 П
51 3	186 ° 187 »	316 Ĩ 317 ĩ	435 Y 436 y	629 e	718 ˘	828_829˘	912 ı	1056 Р
52 4	188 ¼	318 Ĩ 319 ĩ	437 Z 438 z	630 œ	719 ˘	830˘831˘	913 A	1057 С
53 5	189 ½	320 Ĩ 321 ĩ	439 Z 440 Z	631 œ	720 :	832˘833˘	914 B	1058 Т
54 6	190 ¾ 191 ˘	322 Ĩ 323 ĩ	441 Z 442 Z	632 φ	721 ˘	834~835˘	915 Γ	1059 У
55 7	192 Å	324 Ĩ 325 ĩ	443 Z 444 Z	633 ı 634 ı	722 ˘	836˘837˘	916 Δ	1060 Ф
56 8	193 Á	326 Ĩ 327 ĩ	445 Z 446 Z	635 ı 636 ı	723 ˘	838˘839˘	917 E	1061 X
57 9	194 Â	328 Ĩ 329 ĩ	447 p 448 l	637 ı 638 ı	724 ˘	840_841_	918 Z	1062 Ц
58 :	195 Ă	330 Ĩ 331 ĩ	449 l 450 †	639 ı 640 R	725 ˘	842~843˘	919 H	1063 Ч
59 ;	196 Ä	332 Ĩ 333 ĩ	451 !	641 R	726 ˘	844~845˘	920 Θ	1064 Ш
60 <	197 Å	334 Ĩ 335 ĩ	452 DŽ	642 R 643 J	727 -	846_847_	921 I	1065 Ш

Козьминых Н.М.

Строки (string)

5

- Строка - набор символов, заключенный в двойные кавычки
- Строки являются объектами, относятся к ссылочным типам
- Объявление:
 - 1) `string str = "Пример строки";`
 - 2) `char[] charArray = {'e', 'x', 'a', 'm', 'p', 'l', 'e'};`
`string str = new string(charArray);`
- В `System.String` предоставляется набор методов для определения длины символьных данных, поиска подстроки в текущей строке, преобразования символов из верхнего регистра в нижний и наоборот, и т.д.

Методы работы со строками

6

Метод	Назначение
Compare()	Статический метод, который позволяет сравнить две строки
Concat()	Комбинирует отдельные экземпляры строк в одну строку (конкатенация)
Contains()	Метод, который позволяет определить, содержится ли в строке определенная подстрока
Equals()	Метод, который позволяет проверить, содержатся ли в двух строковых объектах идентичные символьные данные
IndexOf()	Находит первое вхождение заданной подстроки или символа в строке
IndexOfAny()	Находит первое вхождение в строку любого символа из набора
Insert()	Метод, который позволяет вставить строку внутрь другой определенной строки
Join()	Строит новую строку, комбинируя содержимое массива строк

Методы работы со строками

7

Метод	Назначение
LastIndexOf()	То же, что IndexOf, но находит последнее вхождение
PadLeft() PadRight()	Методы, которые позволяют дополнить строку какими-то символами, соответственно, справа или слева
Remove() Replace()	Методы, которые позволяют получить копию строки с соответствующими изменениями (удалением или заменой)
Split()	Метод, возвращающий массив string с подстроками, которые отделяются друг от друга элементами, например, пробелами
Substring()	Извлекает подстроку, начиная с определенной позиции строки
ToUpper () ToLower()	Методы, которые позволяют создавать копию текущей строки в формате, соответственно, верхнего или нижнего регистра
Trim()	Метод, который позволяет удалять все вхождения определенного набора символов с начала и конца текущей строки

Пример

```
Строки s1 и s2 не равны
При этом в них есть одинаковый текст
Один, два три, четыре
Слово "это" найдено в строке, оно находится на: 0 позиции
Последнее вхождение слова "это" находится на 13 позиции
```

8

```
// Сравним первые две строки
string s1 = "это строка";
string s2 = "это текст, а это строка";
if (String.CompareOrdinal(s1, s2) != 0)
    listBox9.Items.Add("Строки s1 и s2 не равны");
if (String.Compare(s1, 0, s2, 13, 10, true) == 0)
    listBox9.Items.Add("При этом в них есть одинаковый текст");

// Конкатенация строк
listBox9.Items.Add(String.Concat("\n" + "Один, два ", "три,
    четыре"));

// Поиск в строке - Первое вхождение подстроки
if (s2.IndexOf("это") != -1)
    listBox9.Items.Add("Слово \"это\" найдено в строке, оно" +
        "находится на: " + s1.IndexOf("это") + " позиции");
```

Пример

9

Один из символов из массива ch найден в текущей строке на позиции 1
Подстрока найдена!
Ваша операционная система Windows 7

```
// Последнее вхождение подстроки
if (s2.LastIndexOf("это") != -1)
    listBox9.Items.Add("Последнее вхождение слова \"это\" находится"
        + " на " + s2.LastIndexOf("это") + " позиции");

// Поиск из массива символов
char[] myCh = { 'ы', 'х', 'т' };
if (s2.IndexOfAny(myCh) != -1)
    listBox9.Items.Add("Один из символов из массива ch " +
        " найден в текущей строке на позиции " + s2.IndexOfAny(myCh));

// Определяем начинается ли строка с заданной подстроки
if (s2.StartsWith("это текст") == true)
    listBox9.Items.Add("Подстрока найдена!");

// Определяем содержится ли в строке подстрока
string myOS = Environment.OSVersion.ToString();
if (myOS.Contains("NT 6.1"))
    listBox9.Items.Add("Ваша операционная система Windows 7");
```

Класс `StringBuilder`

10

- Для строки класса **`String`** выделяется ровно столько памяти, сколько необходимо для ее хранения
- Класс **`StringBuilder`** имеет два главных свойства:
 - ▣ **`Length`**, показывающее длину строки, содержащуюся в объекте в данный момент
 - ▣ **`Capacity`**, указывающее максимальную длину строки, которая может поместиться в выделенную для объекта память
- Класс **`StringBuilder`**:
 - ▣ выделяет больше памяти, чем нужно в данный момент
 - ▣ модификация строки происходит внутри блока памяти – эффективно для добавления подстрок и замену символов
 - ▣ имеет свойство *удваивать* свою емкость, когда происходит переполнение, а новое значение емкости не установлено явно

Методы класса `StringBuilder`

11

Метод	Назначение
<code>Append()</code>	Добавляет строку к текущей строке
<code>AppendFormat()</code>	Добавляет строку, сформированную в соответствии со спецификатором формата
<code>Insert()</code>	Вставляет подстроку в строку
<code>Remove()</code>	Удаляет символ из текущей строки
<code>Replace()</code>	Заменяет все вхождения символа другим символом или вхождения подстроки другой подстрокой
<code>ToString()</code>	Возвращает текущую строку в виде объекта <code>System.String</code> (переопределение метода класса <code>System.Object</code>)

Пример

пример класса
пример класса - добавляемый текст
пример класса - добавляемый текст

12

```
StringBuilder str = new StringBuilder("пример  
класс");  
listBox9.Items.Add(str.ToString());  
str.Append(" - добавляемый текст");  
listBox9.Items.Add(str.ToString());  
  
Random ran = new Random();  
for (int i = 'в'; i >= 'а'; i--)  
    str = str.Replace((char)i, (char)(ran.Next(48, 58)));  
listBox9.Items.Add(str.ToString());  
  
for(int i = 48; i < 58; i++)  
    listBox9.Items.Add((char)i);
```

0
1
2
3
4
5
6
7
8
9

Структура Char

13

- Представляет собой символ кода UTF-16

Метод	Назначение
GetNumericValue()	Преобразует указанный числовой символ Юникода в число двойной точности с плавающей запятой
IsDigit()	Показывает, относится ли указанный символ Юникода к категории десятичных цифр
IsLetter()	Показывает, относится ли указанный знак Юникода к категории букв Юникода
IsLetterOrDigit()	Показывает, относится ли знак в указанной позиции в указанной строке к категории букв или десятичных цифр
IsNumber()	Показывает, относится ли указанный символ Юникода к категории цифр
IsSeparator()	Показывает, относится ли указанный символ Юникода к категории знаков-разделителей
IsPunctuation()	Показывает, является ли указанный символ знаком препинания

Пример

14

```
// все цифры переставить в начало
```

```
string ss = "ad2e57b6", temp;  
int kol = 0;  
for(int i = 0; i < ss.Length; i++){  
    if (Char.IsNumber(ss, i))    {  
        temp = ss.Substring(i, 1);  
        ss = ss.Remove(i, 1);  
        ss = ss.Insert(kol, temp);  
        kol++;  
    }  
}  
textBox1.Text = ss;
```

Пример

15

// удалить слова, длина которых равна заданному значению

```
int z = 3;
string ss = "asd tg dfg h yui hj erf";
string[] arr = ss.Split(' ');
StringBuilder res = new StringBuilder();
foreach(string item in arr){
    if (item.Length != z)
        res.Append(item).Append(' ');
}
textBox1.Text = res.ToString();
```

16

Спасибо за внимание

Вопросы...