第七章 字符串类

主要内容

1. String类
2. StringBuffer类

1 String类

- String类（字符串类）的对象是一经创建便不能变动内容的字符串常量。
- 在前面的程序中我们已经多次使用了字符串常量，"Input a Integer data\n"就是字符串常量。
- 注意区分本章学习的字符串常量与字符常量。

1.1 创建String对象

- Java语言规定字符串常量必须用双引号括起，一个串可以包含字母、数字和各种特殊字符，如+、-、*、/、$等。

- Java的任何字符串常量都是String类的对象，没有明确命名时，Java自动为其创建一个匿名String类的对象，也称为匿名String类的对象。

- 可以用下面的方法创建String类的对象。

```
String c1= "Java";
```

语句创建String类的对象c1，并通过赋值号将匿名String类的对象"Java"赋值给c1引用

String类的对象一经创建，便有一个专门的数据成员来记录它的长度(length)。

1.2 String类的构造方法

构造方法	说明
public String ()	创建一个空字符串对象
public String (String value)	用串对象value创建一个新的字符串对象，value可以是字符串或String类的对象。
public String (char value[])	用字符数组value[ ]来创建字符串对象。

public String (char value[], int offset, int count)	从字符数组value中下标为offset的字符开始，创建有count个字符的串对象。
public String (byte ascii[])	用byte型字符串数组ascii，按缺省的字符编码方案创建串对象。
public String (byte ascii[], int offset, int count)	从字节型数组ascii中下标为offset的字符开始，按缺省的字符编码方案创建count个字符的串对象。
public String (StringBuffer Buffer)	构造一个新的字符串，其值为字符串的当前内容

1.3 String类成员方法

成员方法	说明
public int length()	返回当前串对象的长度
public char charAt(int index)	返回当前串对象下标index处的字符

public int indexof (int ch)	返回当前串内第一个与指定字符ch相同的下标，若找不到，则返回-1.
public int indexOf (String str, int fromIndex)	从当前下标fromIndex处开始搜索，返回第一个与指定字符串str相同的第一个字母在当前串中的下标
public String substring (int beginIndex)	返回当前串中从下标beginIndex开始到串尾的子串
public String substring (int beginindex, int endIndex)	返回当前串中从下标beginIndex开始到下标endIndex-1的子串

public boolean equals (Object obj)	当且仅当obj不为null且当前串对象与obj有相同的字符串时，返回true。
public boolean equalsIgnoreCase (String s)	功能与equal类似，忽略大小写
public int compareTo (String another_s)	比较两字符串的大小

public String concat (String str)	将字符串连接在当前串的尾部
public String replace (char oldCh, char newCh)	将字符串oldCh替换为newChar
public String toLowerCase ()	将大写字符转换为小写字符
public String toUpperCase ()	将小写字符转换为大写字符

public static String valueOf (type variable)	返回变量variable值的字符串形式
public static String valueOf (char[] data, int offset, int count)	返回字符数组data从下标offset开始的count个字符的字符串
public static String valueOf (Object obj)	返回对象obj的字符串
public String toString ()	返回当前字符串

1.4 访问字符串对象

- 用于访问字符串对象的信息常用的成员方法：

1. length(); 返回当前串对象的长度；

2. charAt(int index);
   返回当前串对象下标Index处的字符。

3. indexOf(int ch);
   返回与ch相同字符的下标

4. indexOf(String str, int fromIndex);

从当前fromIndex处开始搜索，返回第一个与制定字符串str相同的第一个字母的下标。

5. subString(int beginIndex);

返回当前传中从beginIndex开始到串尾的子串。

6. subString(int beginIndex, int endIndex);

返回当前串中从beginIndex开始到endIndex-1的子串。

1.5 字符串比较

- 常用的字符串比较的成员方法有：

1) equals();

2) equalsIgnoreCase();

3) compareTo();

1. 当前串对象.equals(模式串对象)

当且仅当当前串对象与模式串对象的长度相等且对应位置的字符(包括大小写)均相同时，返回true，否则flase。

2. 当前串对象.equalsIgnoreCase(模式串对象)

与equals用法相同，但不区分大小写。

3. 当前串对象.compareTo(模式串对象)

比较大小，返回字符串长度之差或第一个不同字符的Unicode码值之差。

1.6 字符串操作

- 字符串操作是指已有的字符串对象产生新的字符串对象。

1) concat();

2) replace();

3) toLowerCase();

4) toUpperCase();

1.7 其它类型数据转换为字符串

- String类的valueOf(参数)成员方法可将参数类型的数据转换成字符串。
- 参数类型(type)包括：boolean, char, int, long, float, double

String valueOf(type variable)

String valueOf(char[] data, int offset, int count)

1.8 main方法的参数

- 在Java应用程序中必须写

```
public static void main(String[ ] args)
```

- main方法中有一个参数是字符串数组args，这个数组的元素args[0],args[1].....args[n]的值都是字符串。

- args就是命令行的参数。在Java解释器解释用户的字节码文件时，可以包括需要传给main方法的参数。形式如下：

```
java 类文件名 字符串1 字符串2 ... 字符串n
```

【例7_1】运行时需要输入参数的main方法。

```
public class e7_1
{
   public static void main(String[ ] args)
   {
       for(int i=0; i<args.length; i++)
           System.out.println(args[i]);
   }
}
```

```
Hello
World
Let's
Java!
```

运行时输入"java c7_7 Hello World Let's Java!"命令

2 StringBuffer类

- StringBuffer类是java.lang.Object的子类。与String类不同，StringBuffer类可在操作中更改内容的字符串类。

- StringBuffer类的对象不仅能进行查找和比较，也可做添加、插入、修改之类的操作。

2.1 创建StringBuffer对象

构造方法	功能说明
public StringBuffer ()	创建空字符串缓冲区，默认为16个字符
public StringBuffer (int length)	用length指定初始长度的空字符缓冲区
public StringBuffer (String str)	用指定str创建字符缓冲区，其长度为str的长度再加16个字符

2.2 StringBuffer类的成员方法

成员方法	功能说明
public int length ()	返回当前缓冲区字符串的长度
public char charAt (int index)	返回当前缓冲区字符串下标Index的字符
public void setcharAt (int index, char ch)	将index处的字符改变成字符ch的值
public int capacity ()	返回当前缓冲区的长度

public StringBuffer append (Object obj)	将obj.toString返回的字符串添加到字符串的末尾
public StringBuffer append (type variable)	将变量值转换成字符串再添加到字符串末尾
public StringBuffer append (char[] str, int offset, int len)	将数组从offset开始的len个字符依次添加到末尾

public StringBuffer insert (int offset, Object obj)	将obj.toString返回的字符串插入到offset处
public StringBuffer insert (int offset, type variable)	将变量值转换成字符串，插入到offset处
public String toString ()	将可变字符串转化为不可变字符串

2.3 StringBuffer长度

- StringBuffer类提供成员方法测试缓冲区长度。

1) length();

2) charAt();

3) capacity();

2.4 StringBuffer类append()方法

- StringBuffer类提供字符串添加到当前字符串后面的成员方法：

1) append(Object obj);

2) append(type variable);

3) append(char[]str, int offset, int len);

2.5 StringBuffer类insert()方法

- StringBuffer类提供用于插入字符串到当前字符串中的成员方法：

1) insert(int offset, Object obj);

2) insert(int offset, type variable);

2.6 StringBuffer类setcharAt()方法

- StringBuffer类提供

```
setcharAt(int index, char ch);
```

- 方法将当前字符串下标Index处的字符改变成字符ch的值。