****常用对象****

****1.字节和字符****

字节（Byte）：字节是通过网络传输信息（或在硬盘或内存中存储信息）的单位。字节是计算机信息技术用于计量存储容量和传输容量的一种计量单位，1个字节等于8位二进制，它是一个8位的二进制数，是一个很具体的存储空间。

字符：人们使用的记号，抽象意义上的一个符号。 '1'， '中'， 'a'， '$'， '￥'， ……

谈到字符就不得不提ANSI及UNICODE两种不同的编码方式标准（对这两种编码方式标准在此我只简单提一下，如果大家有兴趣可以自己去查一下），ANSI中的字符采用8bit，而UNICODE中的字符采用16bit。（对于字符来说ANSI以单字节存放英文字符，以双字节存放中文等字符,而Unicode下，英文和中文的字符都以双字节存放）Unicode码也是一种国际标准编码，采用二个字节编码，与ANSI码不兼容。     ANSI规定：一个小于127的字符的意义与原来相同，但两个大于127的字符连在一起时，就表示一个汉字，前面的一个字节（他称之为高字节）从0xA1用到 0xF7，后面一个字节（低字节）从0xA1到0xFE，这样我们就可以组合出大约7000多个简体汉字了。在这些编码里，我们还把数学符号、罗马希腊的字母、日文的假名们都编进去了，连在 ASCII 里本来就有的数字、标点、字母都统统重新编了两个字节长的编码，这就是常说的"全角"字符，而原来在127号以下的那些就叫"半角"字符了。从 UNICODE 开始，无论是半角的英文字母，还是全角的汉字，它们都是统一的"一个字符"！同时，也都是统一的"两个字节"。

在Java当中字符是采用Unicode编码标准的，所以“学Java”这个字符串在Java语言中占10个字节。

1. ****String、StringBuffer和StringBuilder****

String类是final字符串对象一旦被初始化就不会被改变。

String s 1= "abc";

String s 2= "abc";

String s 3= new String("abc");

 s1 = "nba";

"abc"字符串对象并没有被改变，存放在占内存的字符串常量池中，只是引用变量s1指向了新创建的字符串对象"nba"。

s2创建的时候，"abc"已经存在于字符串常量池中，故引用变量s2直接指向了已经存在的"abc"字符串对象，故s1==s2。

 s1创建后，是在字符串常量池中创建了一个"abc"字符串对象。而s2是在堆内存中创建了另外一个"abc"字符串对象。所以，两个对象不是同一个对象。

 String类复写了Object中的equals方法，建立了String类自己的判断字符串对象是否相同的依据。只比较字符串内容，不比较地址，==会比较地址。

String类部分方法

 int length(); 获取字符串中字符的个数（长度）

 char charAt(int index);根据位置获取字符

 int indexOf(int ch);根据字符获取在字符串中的位置，indexO方法参数类型为int是为了既可以支持字符，也可以支持字符在ASCII码中对应的数字。

 int lastIndexOf(int ch);根据字符串获取在字符串中第一次出现的位置。

String[] split(String regex)；将字符串变成字符串数组（字符串的切割），涉及到正则表达式。

 char[] toCharArray();将字符串变成字符数组

 char[] getBytes();将字符串变成字节数组

 String toUpperCase();将字符串中的字母转成大写  
 String toLowerCase();将字符串中的字母转成小写

String replace(char oldCh,char newCh);将字符串中的内容进行替换

 String trim();去除字符串两端空格

String concat(String str);将字符串进行连接

boolean equals(Object obj);  
 boolean equalsIgnoreCase(String str);忽略大小写比较字符串内容。

boolean contains(String str);字符串中是否包含指定字符串

 boolean startsWith(String str);字符串是否以指定字符串开头  
 boolean endsWith(String str);字符串是否以指定字符串结尾

 int compareTo(String str);如果参数字符串等于此字符串，则返回值0；如果此字符串按字典顺序小于字符串参数，则返回一个小于0的值；如果此字符串按字典顺序大于字符串参数，则返回一个大于0的值。

String intern();返回字符串对象的规范化表示形式  
 当调用intern方法时，如果池已经包含一个等于此String对象的字符串（用equals(Object)方法确定），则返回池中的字符串。否则，将此String对象添加到池中，并返回此String对象的引用。

  StringBuffer：就是字符串缓冲区，用于存储数据的容器。  
    1. 长度是可变的。  
    2. 可以存储不同类型数据。  
    3. 最终要转成字符串进行使用。  
    StringBuffer的字符串缓冲区初始容量为16个字符，其实质还是数组。

StringBuffer append(data);  
 StringBuffer insert(index,data);

StringBuffer delete(int start,int end);包含头，不包含尾。  
    StringBuffer deleteCharAt(int index):删除指定位置的元素。

char charAt(int index);  
    int indexO(String str);  
    int lastIndexOf(String str);

  StringBuffer replace(int start,int end,String str);  
    void setCharAt(int index,char ch);  
 StringBuilder

StringBuffer是线程同步的，通常用于多线程。  
    StringBuilder是线程不同步的，通常用于单线程，它的出现能够提高程序效率。

****3.基本数据类型对象包装类****

为了方便操作基本数据类型值，将其封装成了对象，在对象中定义了属性和行为丰富了该数据的操作。

用于描述该对象的类就称为基本数据类型对象包装类。  
    byte       Byte  
    short      Short  
    int          Integer  
    long       Long  
    float       Float  
    double    Double  
    char       Character  
    boolean Boolean

该包装对象主要用于基本类型和字符串之间的转换，自动装箱与拆箱操作：  
    基本类型-->字符串  
    1. 基本类型数值+""  
    2. 用String类中的静态方法valueOf(基本类型数值);  
  
    字符串-->基本类型  
    1. 使用包装类中的静态方法xxx parseXxx("xxx");  
        int parseInt(String str);  
        long parseLong(String str);  
        boolean parseBoolean(String str);  
        只有Character没有parse方法。  
    2. 如果字符串被Integer进行对象的封装，可使用另一个非静态的方法，intValue。  
        将一个Integer对象转成基本数据类型值。  
    3. 用Integer的静态方法valueOf(String s);