二、字符串

字符串初始化后不能改变，在常量池中。

方法区：方法数据、静态数据、常量池

1、 String 类

对字符串进行对象的封装

注意：==和 equals 的区别

例：String s1=”abc”;

Stirng s2=new String(“abc”);

s1==s2;(false)　　s1.equals(s2);(true)

“==”比较引用数据类型时是地址值、也就是对象，比较基本数据类型是值相同

“equals()”只比较引用数据类型

在 String 类中判断字符串内容相同，复写 Object 类中的 equals()

int hasCode()也是复写 Object 类中的 hashCode()

在 Object 类中比的是地址值

1.1、 获取

(1) int　length()　字符串长度

(2) int　indexOf(int ch)　　ch 在字符串中第一次出现位置(ch 是 ASCII 码)

(3) char charAt(int index)　　index 位置上的字符

(4) String　substring(int beginIndex, int beginIndex)　　截取子串

1.2、 判断

(1) boolean　startsWith(str)

(2) boolean　endsWith(str)

(3) boolean　containsWith(str)

(4) boolean　isEmpty()

1.3、 转换

(1) 字符数组——>字符串

构造函数 String (char[])

String (char[],offset,count)

静态方法

S.copyValueOf(char[])

(2) 字节数组——>字符串

构造函数 String (byte[])

String (byte[],offset,count)

静态方法

S.copyValueOf(byte[])

(3) 字符串——>字符数组

char[]　toCharArray()

(4) 字符串——>字节数组

byte[]　getBytes()

(5) 基本数据——>字符串

S.valueOf(基本数据)

1.4、 比较

int compareTo(str)　　小返回负数,等返回 0,大返回正数

2、 StringBuffer 类(线程同步)

构造一个其中不带字符的字符串缓冲区，初始容量为 16 个字符。

特点：是一个容器

是可变长度的

可以对字符串内容修改

缓冲区中可存任意类型的数据

最终需要变成字符串

2.1 添加

StringBuffer append(data) 在缓冲区尾部加数据

StringBuffer insert(index,data) 在指定位置加数据

2.2 删除

StringBuffer delete(start,end) 删除 start 到 end-1 范围的元素

StringBuffer deleteChartAt(index) 删除指定位置元素

2.3 修改

StringBuffer replace(start,end,str) 将 start 到 end-1 替换成 str

StringBuffer setCharAt(index,char) 替换 index 处的字符

2.4 查找(未找到返回-1)

int indexOf(str,fromIndex) 从指定位置查找字符串

int lastIndexOf(str,fromIndex) 从指定位置反向查找字符串

2.5 获子串

String substring(start,end) 获取 start 到 end-1 的子串

2.6 反转

StringBuilder reverse() 字符串反转

3、 StringBuilder 类(线程非同步)

JDK1.5 出现的，方法与 StringBuffer 一样，用于单线程

4、 正则表达式

专门操作字符串，简化书写

组：小括号()表示，每次定义一个小括号就是一组，自动从 1 编号

\n 捕获 n 组；拿前规则第一组”$1”

\d 数字 \D 非数字 \w 全部 .任意

？一次或无 *零或多次 +一次或多次 {I,j}i 到 j 次

1.1 匹配 (String 类中 matches(regex))

例：5-15 位的 QQ 号码校验，0 不开头，只能数字

```
String s=”1345345473”;
String reg=”[1-9][1-9]{4,14}”;
boolean b=s.matches(reg);
```

1.2 切割 (String 类中 split (regex))

例：按“.”切割字符串

```
String reg=”\\.”;
String[] arr=str.split(reg);
```

1.3 替换 (String 类中 replaceAll(regex))

例：我我我来来了了了了——>我来的

```
String s=””
```

String reg=”(.)\\1+”;
String s1=s.replaceAll(reg,”$1”)

1.4 获取

(1) 正则表达式封装成对象
(2) 对象关联要操作的字符串
(3) 获取正则表达式引擎
(4) 通过引擎对符合规则子串进行操作

例：

String reg=” [\w+@\w+\\.\w]”
Pattern p=Parttern.compile(reg)
Matcher m=p.matcher(str);　　（m.matches(str)）
boolean b=m.find();
String s=m.group();