# 1.Object

## Object Class

Object是Java中的顶级父类，Java中所有的类都默认直接或者间接的继承object。Object是Java中唯一没有父类的类，任何一个类的对象都可以用Object对象接住。

## 构造方法摘要

构造方法中提供的所有构造方法默认是public修饰的。Object();

object(){};

## 重要方法

**1.Clone();**

克隆对象。克隆完成之后会产生一个新的对象，这个新对象和原对象的地址不同但是属性值是一样的。

一个对象要想被克隆，那么这个对象对应的类必须实现Cloneable接口，Cloneable接口中没有任何的方法和属性，仅仅用于标识这个类产生的对象可以被克隆。

public class T1 implements Cloneable {

int i;

public static void main(String[] args) throws CloneNotSupportedException {

T1 t = new T1();

t.i = 5;

T1 t1 = (T1) t.clone();

System.out.println(t);

System.out.println(t1);

}

}

**2.equals(Object o);!!!**

判断两个对象是否一致。默认比较的是对象的地址，即使用的是==运算符，在开发中要求能够手动重写。Eclipse中右键Sourec中有自动生成。

**书写步骤：**

1. 判断地址是否一样，即直接使用==运算符。
2. 判断对象是否为空。
3. 判断对象的创建类型是否一致。
4. 判断属性值是否一致。判断之前要将对象强转为顶级父类。判断属性值的时候注意String类型的判断。

Public boolean equals(Object obj){

if (this == obj) //1.判断地址是否一致

return true;

if (obj == null) //2.判断参数是否为空

return false;

if (getClass() != obj.getClass()) //3.判断类型是否一致

return false;

Person other = (Person) obj; //4.强制转换类别

if (age != other.age) //5.判断属性是否一致

return false;

if (gender != other.gender)

return false;

if (name == null) {

if (other.name != null)

return false;

} else if (!name.equals(other.name))

return false;

return true;

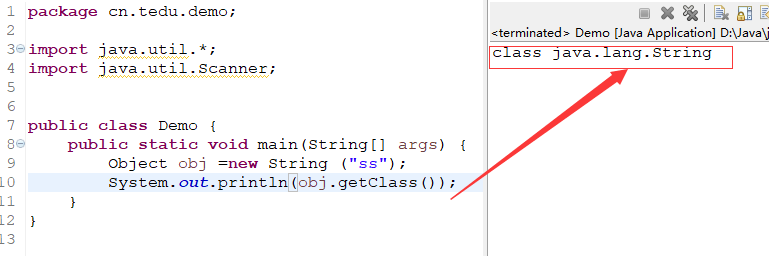
}

**3.finalize();**

通知gc回收垃圾对象，但是垃圾回收器不一定执行。即使gc开始运行，也只回收当前对象而不回收其他对象。

**4.GetClass();!!!**

获取对象的实际类型（创建类型）。



**5.hashCode();**

返回对象的哈希码值的十进制。哈希码是一串32位的二进制数据，由于出现相同的数值的概率非常低，所以可以认为是唯一的。（不建议手动重写）

重写hashCode的时候，当equals值为true的时候，哈希码必须一致。但是哈希码一致equals不一定为true。

native关键字修饰的是本地方法，底层是用c语言写的。

**6.toString();**

对象的字符串形式，默认返回对象的地址。当打印对象的时候默认是调用了这个对象的toString方法。一般会重写。

面试总结：==和equals有什么区别？

当比较基本类型的时候，==判断的是实际数据；对于引用类型而言==判断的是地址。

equals只能作用于引用类型，默认比较的是两个对象的地址；当重写equals的时候按照重写之后的要求进行比较

# 2.String---final String Class

## 2.1特点：

1.String是一个最终类，代表字符串的一个类，所有的字符串都是String的对象。

2.字符串是一个常量；字符串可以被共享。

3.字符串存储在方法区。

4.注意不同的方式创建的对象的个数。有几个对象，就看内存中有几个存储的是地址。

String s = “abc”; --- 1

String s = new String(“abc”); --- 2

String s = “a”; s += “b”; --- 5

String s = “a”; --- 1

s = new StringBuilder(“a”).append(“b”).toString(); --- 4

String s = “a”; s += “b”; //1个a，1个b，2个对象。

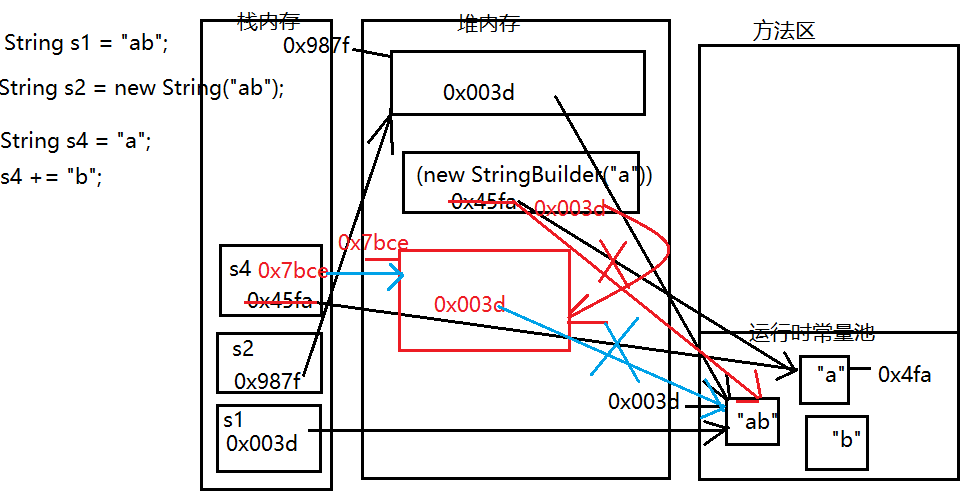
-> String s = “a”; //

-> s = new StringBuilder(“a”).append(“b”).toString();

//3个对象：StringBuilder1个，append拼接产生1个，toString产生1个。

-> s = new String(“ab”);//

最终只存活了4个对象，StringBuilder产生的对象会被解析。



String中“+”拼接字符创建对象的个数，和StringBuiler的append对比？

// 100个字符串组成的字符数组

String[] arr = new String[100];

// + 进行拼接

String s = “”; // 1

for(String str : arr){

s += str; //s = new StringBuilder(s).append(str).toString(); //300

}

// StringBuilder

StringBuilder sb = new StringBuilder(); // 1

for(String str : arr){

sb.append(str); // 100

}

String s = sb.toString(); // 1

在拼接多个字符串的时候最好使用StringBuilder,如果拼接的字符串个数比较少，可以使用+进行拼接

## 2.2 String的构造方法。

#### String();

创建一个字符串对象。

String s = null; //没有对象产生。

String s = “ ”; //有1个对象，存储在常量池中，值为空即null。

String s = new String(); //堆内存中存储一个对象，值为null。

String s = new String(“ ”); //2个对象一个堆内存中，一个方法区中。

#### String(byte[] bs);

将字节数组按照系统默认编码转化为字符串。

byte[] bs = {97,98,100};

String s = new String(bs);

Byte[] bs = “汉字”.getByte(); //转化为字节数组。

String s = new String(bs);

#### String(byte[] bs, Charset ch);

将字符串按照指定的编码编译。

#### String(byte[] bs,int index,int size,Charset ch);

将字节数组按照指定的编码ch，从下标index开始，转化size个。

#### String(char[] ch);

将字符数组中的元素拼接成一个字符串。

char[] cs = { 'a', 'd', 'e', 'g', 'b', 'm', 'o' };

String s = new String(cs);

#### String(char[] ch, int i, int k)

表示从字符数组的指定下标i开始将指定的个数k拼成字符串。

String s = new String(cs, 2, 4);

#### String(int[] codePoints,int offset,int count);

#### String(StringBuffer buffer);

分配一个新的字符串，它包含字符串缓冲区参数中当前包含的字符序列。该字符串缓冲区的内容已被复制，后续对它的修改不会影响新创建的字符串。

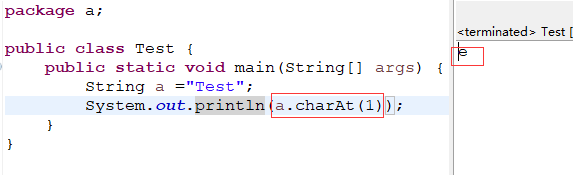
#### String(StringBuilder builder);

分配一个新的字符串，它包含字符串缓冲区参数中当前包含的字符序列。该字符串缓冲区的内容已被复制，后续对它的修改不会影响新创建的字符串。

## 2.3重要的方法

#### charAt(int index)

表示获取这个字符串指定下标index位置上的字符。



#### length()

获取字符串的长度。

注意数组中的length是一个属性，字符串的length();是个方法。

System.out.println(s.length());

练习:统计字符串中字母和数字的个数

输入一个字符串，分别统计之歌字符串中字母和数字的个数。

Scanner s = new Scanner(System.in);

String str = s.nextLine();

s.close();

int letterNo=0,numberNo=0;

for (int i = 0; i < str.length(); i++) {

if(str.charAt(i)>='a'&&str.charAt(i)<='z'||str.charAt(i)>='A'&&str.charAt(i)<='Z'){

letterNo++;

}else if(str.charAt(i)>='0'&&str.charAt(i)<='9'){

numberNo++;

}

}

System.out.println("字母为"+letterNo+"个,数字为"+numberNo+"个");