复习：

1. abstract

抽象类：不一定包含抽象方法，不可以实例化，有构造方法。子类在继承了抽象类之后要实现抽象类中所有的抽象方法。抽象类一定不能用final修饰

抽象方法：没有方法体。抽象方法可以重载，不能用static/final/private修饰

2. interface：用implements让类实现接口。---多实现，接口之间是多继承。 --- 一个类在实现了接口之后需要实现接口中的所有的抽象方法。接口中只允许存在抽象方法，默认用public abstract修饰。 --- 接口中的属性默认用public static final修饰

3. 内部类：方法内部类、成员内部类、静态内部类、匿名内部类

除了静态内部类以外，其余的内部类都只能定义静态常量，不能定义静态变量和静态方法；其余的内部类可以使用外部类中的非静态属性和非静态方法

方法内部类如果使用当前方法中的数据，要求这个数据是一个常量

匿名内部类实际上是继承了对应的类或者是实现了对应的接口

4. 包：package，import --- \* 表示导入当前包下的所有类，但是不包括子包下的类

5. 垃圾分代回收机制：针对堆内存。

API --- Application Programming Interfaces --- 应用程序接口

## Object

是Java中所有类的父类---顶级父类

class Person {} --- 默认继承Object

### 重要方法

clone() --- 克隆出一个新的对象，新对象的属性值和原来的对象一样。如果一个类的对象想要被克隆，那么这个类必须实现一个接口---Cloneable ---这个接口中没有任何的属性和方法，仅仅用于标志

finalize() --- 通知垃圾回收器进行回收，但是GC不一定会执行

练习：重写Student类中的equals

class Student {

private String name;

private String no;

}

## String

代表字符串的类。本身是一个最终类。

字符串在Java底层实际上是以字符数组形式来存储的。

字符串是常量，字符串是被共享的。

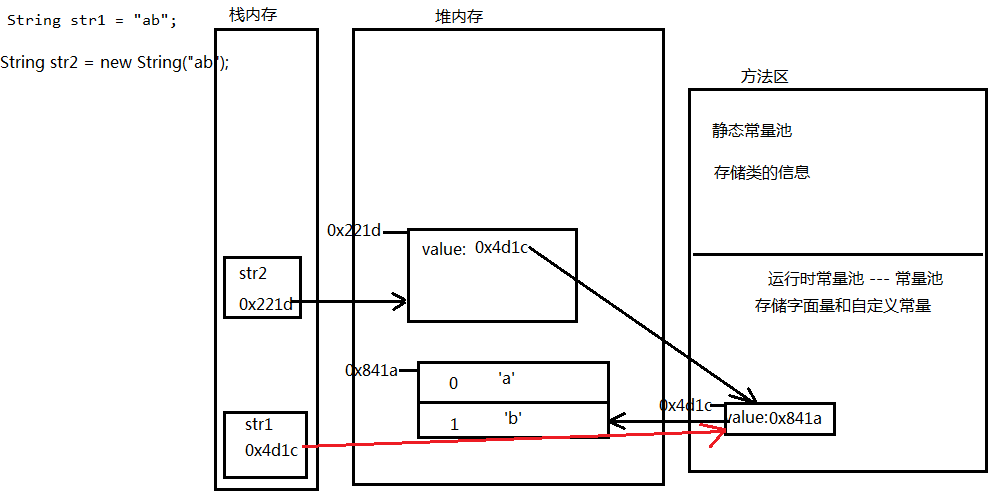
String str = “abc”; --- 1

String str2 = new String(“abc”); --- 2

String str3 = “a”; str3 += “b”; --- 5

String str3 = “a”;

str3 = new StringBuilder(str3).append(“b”).toString(); --- 4



String[] arr = {100个元素};

// 将字符串数组中的元素拼接成一个字符串

String str = “”; // 1

for(String s : arr){

str = str + s; // 每拼接1次，要额外产生3个对象。拼接100次，额外产生300个对象

}

// 整个过程中，一共产生了301个对象

StringBuilder sb = new StringBuilder(); // 1

for(String s : arr){

sb.append(s); // 每拼接1次，要额外产生1个对象。拼接100次，额外产生100个对象

}

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

// 整个过程中，一共产生了102个对象

在拼接字符串的时候，如果个数比较少可以使用+;如果个数比较多，建议使用StringBuilder

StringBuilder是线程不安全的类，StringBuffer是线程安全的。

练习：

1. 输入一个字符串，统计字符串中字母、数字以及其他符号的个数

2. 输入一个字符串，提取这个字符串中的数字并且求和 - dhda3bfkld5nl1ln6 -> 3 5 1 6 -> 15

3. 输入一个字符串，提取这个字符串中的数字然后进行升序排序 - dhda3bfkld5nl1ln6 -> 3 5 1 6 -> 1356

4. 输入一个字符串，输出每一个字符出现的次数 - sahlgaaslfn - 标记

s:2 a:3 h:1 l:2 g:1 f:1 n:1