复习：

1. abstract：

抽象类：不能实例化，但是类中有构造方法。子类继承了抽象类之后必须重写其中的抽象方法。抽象类不能用final修饰

抽象方法：没有方法体。必须在子类中被重写。可以重载。不能用static/final/private修饰

2. interface：

类和接口之间用implements关键字来完成实现关系。类在实现接口之后需要重写接口中所有的抽象方法。在JDK1.8以前，接口中只能定义抽象方法 --- 默认用public abstract修饰。接口中可以定义属性但是默认使用public static final修饰。从JDK1.8开始，接口中允许定义实体方法，必须用default修饰，也可以定义静态方法。

接口与类之间是多实现，接口之间是多继承

作用：作为模板或者是协议、约束等来使用。

3. 内部类：

方法内部类、成员内部类、静态内部类（嵌套类）、匿名内部类

除了静态内部类以外，其余的内部类中都不允许定义静态变量和静态方法

除了静态内部类以外，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

方法内部类使用当前方法中的数据的时候，要求这个数据得是一个常量。

匿名内部类本质上是实现了对应的接口或者是继承了对应的类。所以任何一个接口都可以产生匿名内部类，任何一个可以被继承的类都可以产生匿名内部类。

4. 包：

声明包用的是package，作用：区分同名类，进行功能/模块的划分

导入包用的是import，给要使用的类指定查询路径的。 --- \* 表示导入当前包下的所有的类但是不包括子包下的类

在使用java.lang包以及同包下的类可以不用导包

5. 垃圾分代回收机制：

针对堆内存。

对象一开始是先放到新生代的伊甸园区，在伊甸园区经过依次回收如果依然存在则挪到幸存区，在幸存区如果经过多次回收依然存在则挪到老生代。在回收时先扫描新生代，如果释放的空间足够则不扫描老生代。

初代回收（minor gc在新生代）

完全回收（full gc 在老生代）

如果一个对象在新生代存放不开，则会试图进行一次回收，如果经过回收依然在新生代存不开，则直接试图放入老生代。如果老生代也存放不开则会报OutOfMemoryError

## Object

顶级父类。任何一个类都将Object作为父类，也就意味着任何一个对象都可以赋值给Object对象。Object类也是Java中唯一的一个没有父类的类。

### 重要方法

clone() --- 如果一个对象要想被克隆，那么这个对象所对应的 类必须实现接口--- Cloneable --- 这个接口中没有任何的方法和属性仅仅起标志性的作用

finalize() --- 通知GC回收垃圾，GC不一定启动

练习：定义一个类表示用户，重写其中的equals方法

class User {

private String username;

private String password;

}

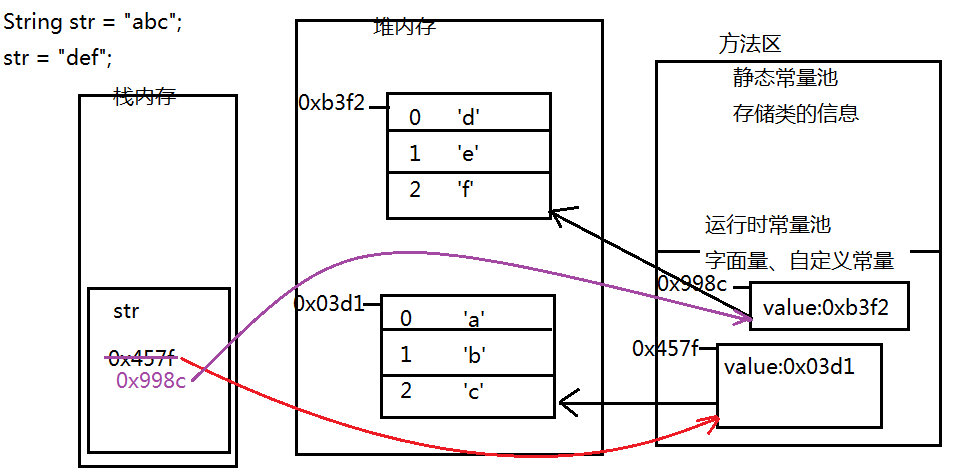
## String

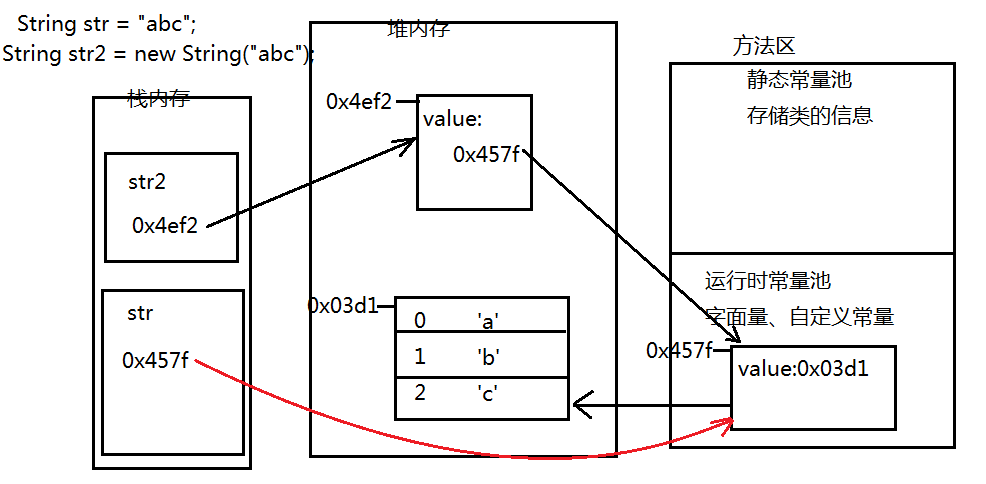
最终类（final）。代表字符串的类，所有的字符串都是String的对象。

字符串是一个常量，定义好之后不可改变。

因为字符串是一个常量，所以它是被共享的 --- 只要值相同，用的都是同一个字符串

字符串本质上是一个不可变的字符数组（final char[]）





String str = “abc”; --- 1

String str = new String(“abc”); --- 2

String str = “a”; --- 1

str = str + “b”; ->

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

String[] strs = {/\* 100个元素 \*/};

// 将字符串数组中的所有元素进行拼接

// 使用+进行拼接 --- 整个过程要产生301个对象

String str = “”; // 1

for(String s : strs)

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

// 使用StringBuilder进行拼接 --- 整个过程中要额外产生102个对象

StringBuilder sb = new StringBuilder(); // 1

for(String s : strs)

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

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

如果拼接的字符串个数比较多，建议使用StringBuilder；如果个数比较少的话可以使用+

练习：

1. 输入字符串，然后再输入2个数字(一个表示起始位置，一个表示结束位置)，按照指定的数字截取一个子字符串

gahodnlasfho 2 5 -> hod

2. 输入一个字符串，统计字符串中字母、数字和其他符号的个数

3. 输入一个字符串，将其中的数字提取出来并求和

dsho353sdhiha89dnso4 -> 3+5+3+8+9+4 = 32

4. 输入一个字符串，将其中的数字提取出来并进行升序排序

dsho353sdhiha89dnso4 -> 353894 -> 334589

5. 输入一个字符串，统计其中每一个字符出现的次数

sdhaofnaadsg -> s:2 d:2 h:1 a:3 o:1 f:1 n:1 g:1