**自我介绍**

String s1 = "coder";

String s2 = "coder";

String s3 = "coder" + s2;

String s4 = "coder" + "coder";

String s5 = s1 + s2;

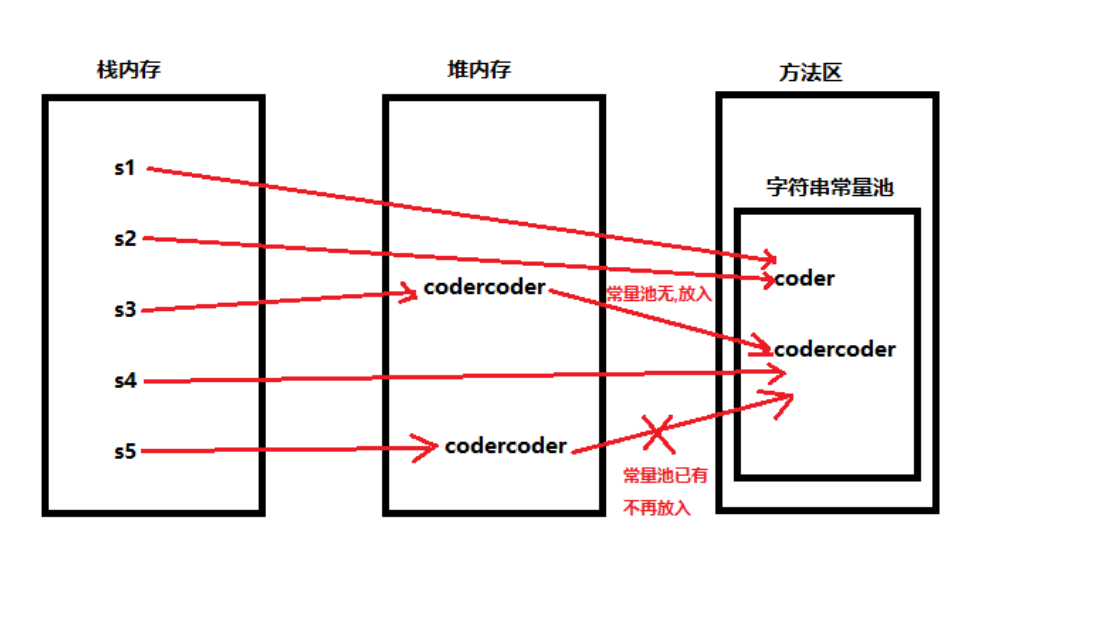
System.out.println(s3 == s4);

System.out.println(s3 == s5);

System.out.println(s4 == "codercoder");

false；false； true；

s=“a”+“b” 是字符串拼接， s = "a"+b 则是重新创建一个对象



import java.util.\* ;

能访问java/util目录下的所有类，不能访问java/util子目录下的所有类

&运算符：两个数都转为二进制，然后从两个数的最高位进行与运算，两个都为真（1），结果才为真（1），否则为假（0）

可以把任何一种数据类型的变量赋给Object类型的变量

byte a1 = 2, a2 = 4, a3;

short s = 16;

a2 = s;

a3 = a1 \* a2;

同级之间转换需要强转，高级向低级转换也需要强转

**int [ ]  x = new int[10] ；**

x[9]为0

编译时异常必须显示处理，运行时异常交给虚拟机。

int i = 0;

i = i++ + i;

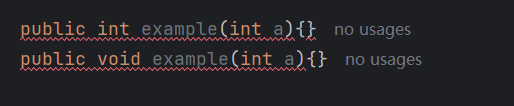
在 i = i++ + i 中，先执行的是 i++ ，此时式子可为 i = 0 + i ，那么之前已经执行了i++之后，i变为1，即i=0+1。

**一个Java源程序文件中定义几个类和接口，则编译该文件后生成几个以.class为后缀的字节码文件**

这个题目是说一个Java源程序中有n个类和接口，就会生成n个以.class为后缀名的文件，但是它忽略了在类和接口里面还可以定义内部类和匿名内部类，这个时候就不仅仅是n个了，还需要加上定义了多个内部类和匿名内部类。

比如说在一个Java源程序中定义了1个接口，1个类，而在这个接口里面定义了一个内部类，在这个类中定义了一个匿名内部类，这样就是不是2个class文件，而是4个class文件

当用javac命令生成编译这个.java 文件的时候，则会针对每一个类生成一个.class文件；



byte b1=1,b2=2,b3,b6,b8;

final byte b4=4,b5=6,b7;

b6=b4+b5;    /\*语句2\*/

被声明final 所以类型是不会转换

被final修饰的变量不会自动的改变类型,当两个final修饰的变量操作时,结果会根据左边变量的类型而转换

system.gc只是提醒，不是强制GC

^表示异或 就是相同是0 不同是1

Vector与ArrayList一样，也是通过数组实现的，不同的是Vector支持线程的同步

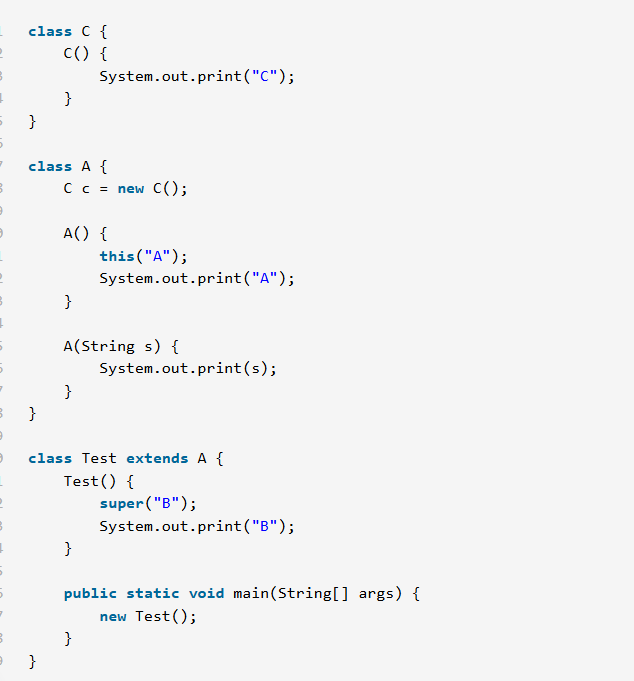
Properties实现了Map接口，是线程安全的

基本数据类型不是对象

如果是等待清理队列中如果又被调用，则不会执行finallize方法

采用高版本的JDK编写的程序，在低版本的JRE中无法运行

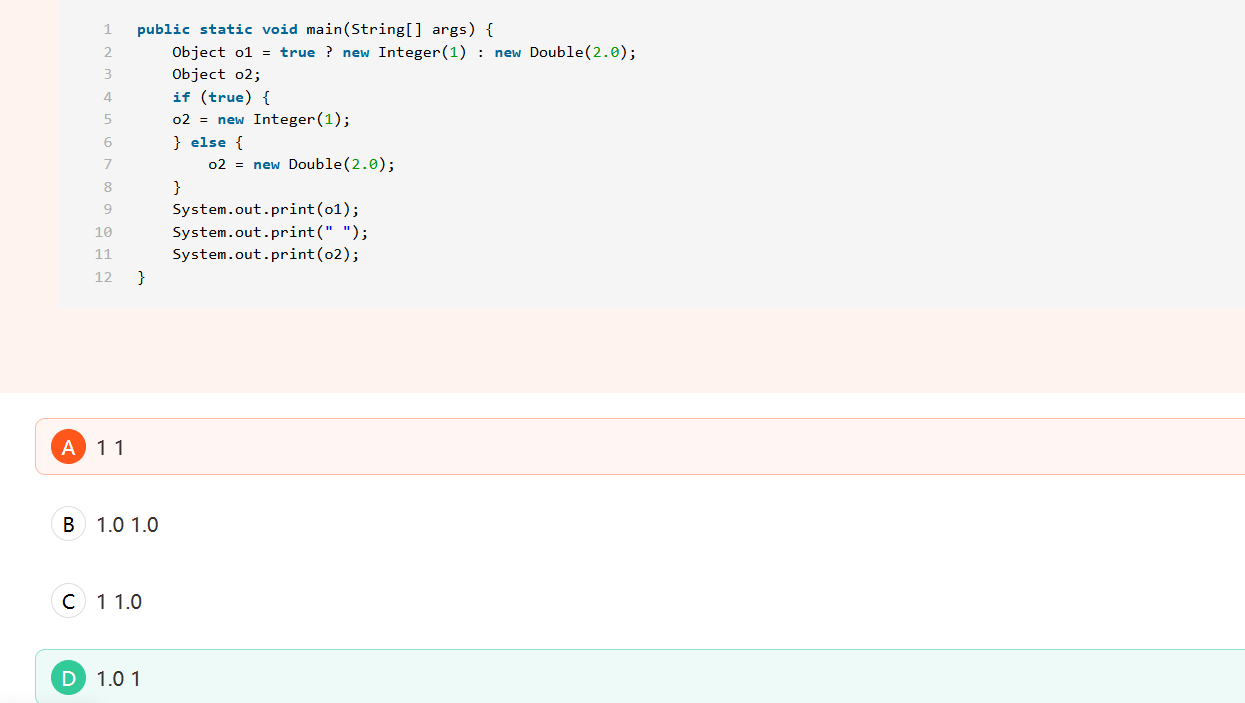
static块只执行一次，在JVM加载类的时候



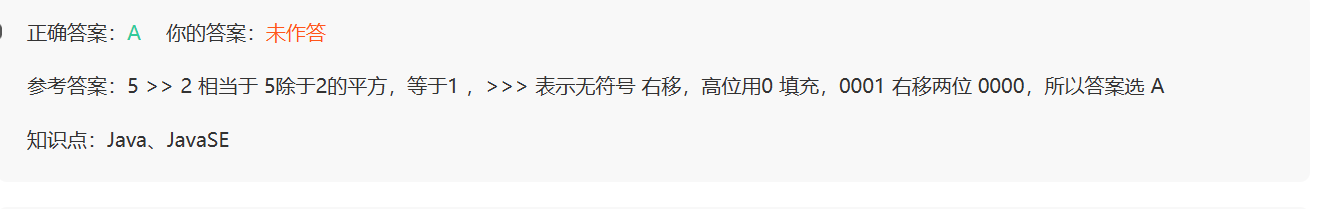
首先new了一个子类对象，那么就要调用构造方法来初始化该子类对象，但是该类继承自A，所以要先调用父类的构造方法，这里通过super("B")显示的调用了父类的带参构造。执行父类的带参构造前要先对父类中的对象进行初始化，对父类中的c成员进行初始化，调用了C类的无参构造  
CBB

char s = ‘\u0639’; 对

三元运算符会对两个结果的数据类型，进行自动的类型提升







int b = 3;  
b = b++;

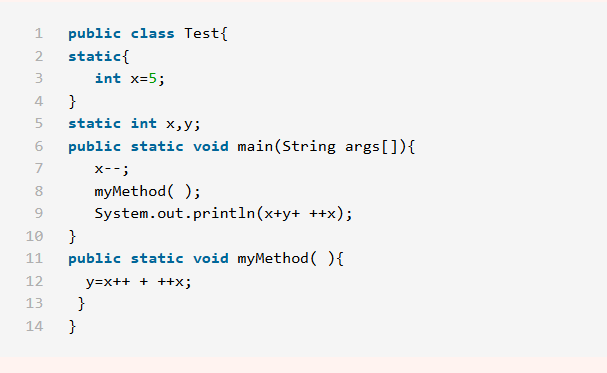
值不变

byte b = 127;  
b++;

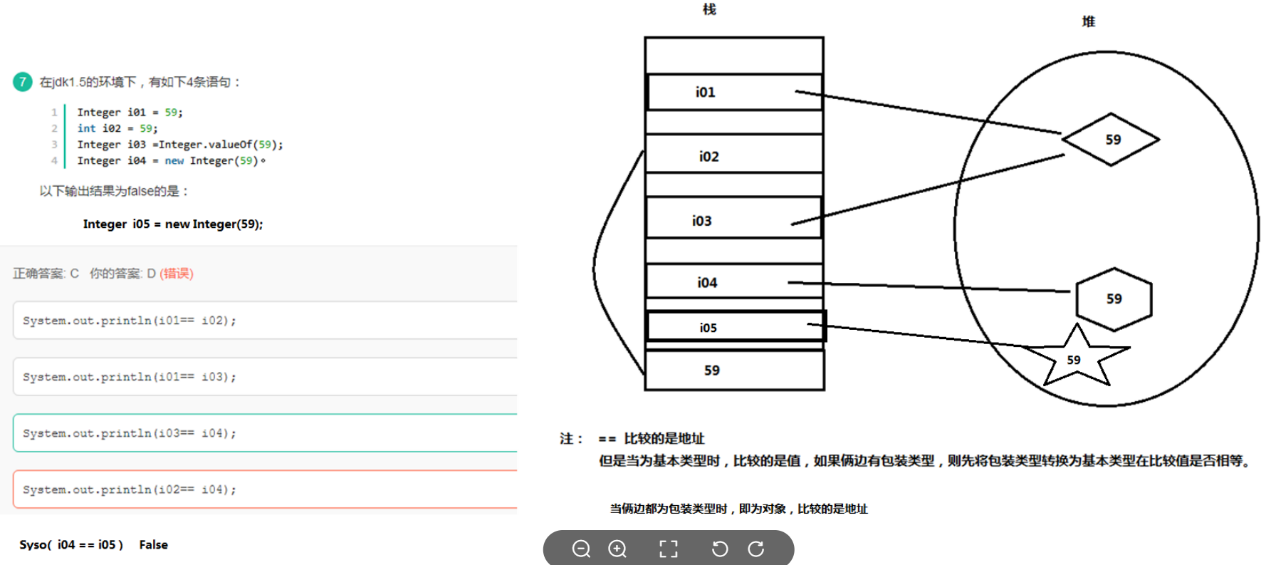
-128

y=x++ + ++x; -1 + 1=0

System.out.println(x+y+ ++x); 1+0+2=3



如果try，finally语句里均有return，忽略try的return，而使用finally的return



RuntimeException 体系包括错误的类型转换、数组越界访问和试图访问空指针等等,必须 被 try{}catch 语句块所捕获 X

注意“必须”, java里面异常允许被抛出, 要记得 throws 关键词的使用

Number类可以被继承

下面这条语句一共创建了多少个对象：String s="welcome"+"to"+360;

1

字面量"+"拼接是在编译期间进行的，拼接后的字符串存放在字符串池中

标识符由26个英文字符大小写（a~z，A~Z）、数字(0~9)、下划线(\_)和美元符号($)组成；

子类能继承父类的所有成员

java语言的下面几种数组复制方法中，哪个效率最高

System.arraycopy

super.getClass().getName()

返回：包名+类名

Double oD=3; X

 父类private的成员变量，根据权限修饰符的访问控制范围，只有在类内部才能被访问，就算是他的子类，也不能访问

final修饰变量，变量的引用（也就是指向的地址）不可变，但是引用的内容可以变（地址中的内容可变）

-n=~n-1

局部变量是方法加载的时候创建的，而不是方法调用的时候创建的

对象空间被收集前执行finalize（）方法

子类引用父类的静态字段，只会触发子类的加载、父类的初始化，不会导致子类初始化 ，而静态代码块在类初始化的时候执行

先分析一下里面各个参数的含义：   
-Xms：1G ， 就是说初始堆大小为1G   
-Xmx：2G ， 就是说最大堆大小为2G   
-Xmn：500M ，就是说年轻代大小是500M（包括一个Eden和两个Survivor）   
-XX:MaxPermSize：64M ， 就是说设置持久代最大值为64M   
-XX:+UseConcMarkSweepGC ， 就是说使用使用CMS内存收集算法   
-XX:SurvivorRatio=3 ， 就是说Eden区与Survivor区的大小比值为3：1：1  
题目中所问的Eden区的大小是指年轻代的大小，直接根据-Xmn：500M和-XX:SurvivorRatio=3可以直接计算得出  
500M\*(3/(3+1+1))

垃圾回收在jvm中优先级相当相当低

long和double都占了64位（64bit）的存储空间。

b = x > 50 && y > 60 || x > 50 && y < -60 || x < -50 && y > 60 || x < -50 && y < -60;

优先级排序是：'>'，'<'，'&&'，'||'。 即 b=(x>50&&y>60)||(x>50&&y<-60)||(x<-50&&y>60)||(x<-50&&y<-60);

在private  修饰不能在外部类中调用，main 方法属于Test类的方法， 所以 对象 t 可以在他自己的类方法中调用它的private

HashTable使用Enumeration，HashMap使用Iterator

成员变量被final修饰，必须显式初始化

join()底层就是调用wait()方法的

构造函数不能被继承，构造方法只能被显式或隐式的调用

1、finally块一定会执行（除非退出虚拟机），无论是否try…catch。

2、finally块前有return，会先执行return语句，并保存下来，再执行finally块，最后return。

3、finally块前有return、finally块中也有return，先执行前面的return，保存下来，再执行finally的return，覆盖之前的结果，并返回