**中软国际考题-笔试**

1. 单选题

1、下列内部静态类声明的正确的是（A）

**public** **class** A {

**static** **class** B{};

}

public static class A {

static class B{};

}

1. public static class A{};
2. public final static class One{};

2、**给出如下代码:**

class Test{

　　private int m;

　　public static void fun() {

　　　　// some code...

　　}

}

如何使成员变量m 被函数fun()直接访问?（C）

A．将private int m 改为protected int m

B．将private int m 改为 public int m

C．将private int m 改为 static int m

D．将private int m 改为 int m

3、在一类中被定义为且用Static修饰的类，称为（A ）。

A．静态成员类 B．匿名类

C．成员类 D．局部类

4、给出下列代码，如何使成员变量m被方法fun()直接访问（C）

class Test{

private int m;

public static void fun(){

}

A. 将private int m改成protected int m

B. 将private int m改成public int m

C. 将private int m改成static int m

D. 将private int m改成 int m

5、阅读下面的代码

class Demo{

privateString name;

Demo(String name){this.name = name;}

privatestatic void show(){

System.out.println(name)

}

publicstatic void main(String[] args){

Demo d = new Demo(“lisa”);

d.show();

}

}

下列关于程序运行结果的描述中，正确的是（D）

A、输出 lisa B、输出null C、输出name

D、编译失败，无法从静态上下文中引用非静态变量name

6、对静态成员（用static修饰的变量或方法）的不正确描述是（C）

A. 静态成员是类的共享成员

B. 静态变量要在定义时就初始化

C. 调用静态方法时要通过类或对象激活

D. 一个静态方法不能访问普通方法

7、成员函数可声明为静态的，条件是它不能访问（C）成员

A．静态

B．公有

C．非静态

D．受保护

8、下列描述中正确的是（B）

A. 静态类与其他类毫无关系

B. 静态类为静态内部类

C. 静态类只能在本类调用

D. 静态类无法被继承

9、下列关于静态类描述错误的是（B）

A. 静态类只能为子类

B. 静态类可以为java文件类

C. 静态内部类调用外部类的构造函数，来构造外部类

D. 外部类需要使用内部类，会考虑采用静态类。

10、java中静态类描述正确的是（D）

A.内部类没有普通类的所有特性

B. Java中不存在静态对象

C. 静态类无法被实例化

D. static不能用于最外层的类

11、静态内部类实例化对象的做法正确的是( B)

    A．OuterClass inner = new InnerClass()

    B. Outer.Inner inner = new Outer.Inner()

   C . Inner inner = new Outer ()

  D . Inner inner = new Inner ()

**12、这个程序会有下面哪种结果？（B）**

**public class Test5 {**

**static int value = 9;**

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

**new Test5().printValue();**

**}**

**private void printValue() {**

**int value = 69;**

**System.out.println(this.value);**

**}**

**}**

**A. 编译错误**

**B. 打印9**

**C. 打印69**

**D. 运行时抛出异常**

**13. 关于以下application的说明，正确的是（ C）**

**class StaticStuff {**

**{**

**static int x=10；**

**static { x+=5；**

**}**

**public static void main（String args[ ]）{**

**System.out.println(“x=” + x);**

**}**

**static {**

**x/=3;**

**}**

**}**

**A、4行与9行不能通过编译，因为缺少方法名和返回类型**

**B、9行不能通过编译，因为只能有一个静态初始化器**

**C、编译通过，执行结果为：x=5**

**D、编译通过，执行结果为：x=3**

**14. 下列代码的编译运行结果是：C**

**class A {**

**static {**

**System.out.print("1");**

**}**

**public A() {**

**System.out.print("2");**

**}**

**}**

**class B extends A{**

**static {**

**System.out.print("a");**

**}**

**public B() {**

**System.out.print("b");**

**}**

**}**

**public class Hello {**

**public static void main(String[] args) {**

**A ab = new B();**

**ab = new B();**

**}**

**}**

**A．2a2b1b。**

**B．2a1b1b**

**C．1a2b2b**

**D．1a2b1b**

**A.编译失败 B.运行异常 C.false D.true**

**15. 编译运行下列代码的结果: A**

**public class Demo1 {**

**private int count;**

**public static int getInstanceCount(){**

**return count;**

**}**

**public Demo1() {**

**count++;**

**}**

**public static void main(String[] args) {**

**new Demo1();**

**new Demo1();**

**new Demo1();**

**System.out.println(Demo1.getInstanceCount());**

**}**

**}**

**A.类Demo1编译失败 B.13行输出3**

**C. 13行输出1 D.当执行第10行时，出现一个运行时异常**

**16. 编译运行下列程序的结果是：C**

**public class Demo1{**

**public static void main(String[] args) {**

**aMethod();**

**}**

**public static void aMethod(){**

**try {**

**System.out.println("try");**

**return;**

**}finally{**

**System.out.println("finally");**

**}**

**}**

**}**

**A. 程序编译失败**

**B. 程序编译成功，但在运行的时候发生异常**

**C. 打印输出try和finally**

**D. 打印输出try**

**E. 打印输出finally**

1. 多选题
2. 下列说正确的是？（A、B，C）
3. 静态内部类可以声明普通成员变量和方法;
4. 静态内部类跟静态方法一样，只能访问静态的成员变量和方法，不能访问非静态的方法和属性
5. 静态内部类声明需要static修饰;
6. 任意一个类都可以使用static修饰，无论是内部类还是外部类;

2、Static特点描述正确的是？（A、B、C、D、E）

A，static是一个修饰符，用于修饰成员。

B，static修饰的成员被所有的对象所共享。

C，static优先于对象存在，因为static的成员随着类的加载就已经存在了。

D，static修饰的成员多了一种调用方式，就可以直接被类名所调用 。 类名.静态成员 。

E，static修饰的数据是共享数据，对象中的存储的是特有数据。

3、Java的内部类分为以下两种内部类（A、B）

普通内部类 B. 静态内部类 C. 非静态内部类 D. 匿名内部类

4、关于静态和非静态说法正确的是（A、C）

1. 静态不能修饰局部变量 B. 静态可以修饰局部变量
2. 静态可以调用非静态方法 D. 静态不可以调用非静态方法

5、下列关于静态方法的描述中，正确的是（ A.B.D ）。

A、静态方法属于类的共享成员

B、静态方法是通过"类名.方法名"的方式来调用

C、静态方法只能被类调用，不能被对象调用

D、静态方法中可以访问静态变量

6、对静态成员（用static修饰的变量或方法）的正确描述是（ABD）

A. 静态成员是类的共享成员

B. 静态变量要在定义时就初始化

C. 调用静态方法时要通过类或对象激活

1. 一个静态方法不能访问普通方法

7、下面哪些判断是正确的是（BD），选择二项

A． static方法中能处理非static域

B．abstract类中能有private方法

C．abstract类中不能包含final修饰的方法

D．private可以和static一起修饰一个方法

8、以下关于final关键字说法错误的是？（A、C）

A. 静态类不一定是内部类

B. 全局类无法定义为静态类

C. 非内部类中的方法无法修饰为静态方法

D. 静态类与静态方法可以分别修饰

9、关于静态内部类的说法正确是（ A D ）

A．静态内部类属于外部类，而不是属于外部类的对象。

B. 生成静态内部类对象的方式 static Inner inner = new Inner()。

C. 不能访问外部类的静态成员变量或者静态方法。

D. 不依赖于外部类的实例,直接实例化内部类对象

10、编译运行下面的程序结果是：【BD】  
**public** **class** Demo1 {

**public** **static** **void** main(String[] args) {

print();

}

**public** **static** **void** print(){

System.*out*.println("Demo1");

}

**public** **void** print(){

System.*out*.println("A nother Demo1");

}

}  
A.代码编译成功，打印输出Demo1  
B.代码编译失败，如果将第8行的方法声明改为public void print(int x),代码就能通过编译。  
C.编码编译失败，因为静态的main方法调用了非静态的print方法  
D.编码编译失败，因为存在同名方法。5行和8行参数相同了  
E.代码编译失败，将第8行方法的返回值改为int就能通过编译。

11. 下列哪两个选项单独地插入到第3行，能使该类编译通过【BD】  
 class Bar{

static void foo(int… x){

//insert code here

}

}

A.foreach(x)System.out.println(z);

B.for(int z:x) System.out.println(z);

c.while(x.hasNext())System.out.println(x.next);

D.for(int i = 0;i<x.length;i++) System.out.println(x[i]);

1. 填空题
2. 如果一个类要被声明为static的，只有一种情况，就是\_\_\_\_\_静态类（或静态内部类或嵌套类）\_\_\_\_\_。如果在外部类声明为static，程序会编译\_\_\_\_不通过（或错误）\_\_\_\_\_
3. 静态方法中\_\_\_\_\_\_\_\_\_\_使用this或者super关键字。（不可以）

3、Java的内部类有两种，一种是普通内部类，另一种是 。（静态内部类）

4、创建静态内部类的对象时，不需要外部类的对象，可以直接创建 内部类 对象名= ; （new 内部类())

5、静态内部类不能直接访问外部类的非静态成员，但可以通过 的方式访问。（new 外部类().成员 ）

1. Static修饰符能修饰\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_。(成员变量、方法)

7、用修饰符 static 说明的成员变量是类变量。

8、静态代码块和构造代码块两者先运行的是\_\_\_\_\_\_\_\_\_\_。(静态代码块)

9、static关键字分别用于修饰、\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_。(变量、方法、类)

10、定义静态类格式： \_\_\_\_\_\_\_\_\_ Test{

\_\_\_\_\_\_\_\_\_ class \_\_\_\_\_\_\_\_\_{}

} 。(class static Test1 )

11、静态内部类需要被static修饰

1. 静态方法不能使用\_\_synchronized\_\_方法。

**13、静态代码块不会优先于\_\_构造代码块 \_\_运行**

1. 判断题
2. 任意类都可以声明为静态类（F）（静态类只能在一个类的内部去什么，因此也称为静态内部类或嵌套了，否则是无法声明的）
3. 要创建嵌套类（静态内部类）的对象，必须要其外围类的对象（F）（要创建嵌套类的对象，并不需要其外围类的对象）
4. 不能从嵌套类的对象中访问非静态的外围类对象（T）
5. 成员变量只能被对象调用。（ T ）
6. 成员变量随着对象的创建而存在，随着对象的被回收而释放。（T）
7. 静态变量随着类的加载而存在，随着类的消失而消失。（T）
8. 非静态内部类的非静态成员不可以访问外部类的非静态变量。（ F ）
9. 静态内部类的非静态成员可以访问外部类的静态变量，而不可访问外部类的非静态变量。（T）
10. 静态内部类只能访问外部类的静态成员变量与静态方法。（T）
11. 静态内部类可以有静态成员，而非静态内部类则不能有静态成员。 （T）
12. 如果类中成员变量被定义为静态，那么无论有多少个对象，静态成员变量不只有一份内存拷贝，即所有对象共享该成员变量（ F ）
13. 将一个类设计为单例设计模式,需要定义一个静态的成员方法将本类创建的对象返回给调用者。（F ）
14. 被static关键字修饰的成员变量被称为静态变量，它可以被该类所有的实例对象共享。（T）
15. 静态方法不能使用synchronized关键字来修饰。（F）
16. static不能修饰方法中定义的变量。（ T ）
17. 静态类无法修饰java文件类。（ Y ）
18. 静态内部类只能够访问外部类的静态成员（ T ）
19. 静态内部类可以有非静态成员( F )
20. 如果一个内部类不是被定义成静态内部类，那么在定义成员变量或者成员方法的时候，是不能够被定义成静态的。（ T ）
21. 静态变量可以被类和对象调用。（ T ）
22. 在有多个静态代码块的情况下，虚拟机会随机运行。（F）

五、简答题

# 1、什么是静态类？静态类又称为什么？

答案：(1)如果不需要内部类对象与其外围类对象之间有联系，那么可以将内部类声明为static。被声明为static的类称为静态类。

(2)如果想把一个类声明为static，那只能在一个类的内部去声明另一个类，否则无法声明，因此称为静态内部类，静态类通常也称为嵌套类。

2、成员变量和静态变量的区别？（5分）

答案：

1，两个变量的生命周期不同。

成员变量随着对象的创建而存在，随着对象的被回收而释放。

静态变量随着类的加载而存在，随着类的消失而消失。

2，调用方式不同。

成员变量只能被对象调用。

静态变量可以被对象调用，还可以被类名调用。

3，别名不同。

成员变量也称为实例变量。

静态变量称为类变量。

4，数据存储位置不同。

成员变量数据存储在堆内存的对象中，所以也叫对象的特有数据.

静态变量数据存储在方法区(共享数据区)的静态区，所以也叫对象的共享数据

3、简述内部类、静态内部类、匿名内部类的区别

答：内部类：

成员内部类可访问外部类所有的方法和成员变量。

不能有静态的方法和成员变量。

静态内部类：

    只能访问外部类的静态成员变量与静态方法。

    静态内部类的非静态成员可访问外部类的静态变量，而不可访问外部类的非静态变量。

匿名内部类：

    没有类名，没有class关键字也没有extends和implements等关键字修饰。

    类的定义和对象的实例化同时进行。

4、简述静态内部类和普通内部类的区别

答：

（1）普通内部类不能声明static的方法和变量

普通内部类不能声明static的方法和变量，注意这里说的是变量，常量（也就是final static修饰的属性）

还是可以的，而静态内部类形似外部类，没有任何限制。

（2）使用静态内部类，多个外部类的对象可以共享同一个内部类的对象。

使用普通内部类，每个外部类的对象都有自己的内部类对象，外部对象之间不能共享内部类的对象。

5、简述实例变量和类变量的区别。

解答：类变量也叫静态变量，也就是在变量前加了static 的变量；实例变量也叫对象变量，即没加static 的变量；

区别在于：类变量是所有对象共有，其中一个对象将它值改变，其他对象得到的就是改变后的结果；而实例变量则属对象私有，某一个对象将其值改变，不影响其他对象；

6、简述实例方法和类方法的区别

解答：实例方法可以对当前对象的实例变量进行操作，也可以对类变量进行操作，实例方法由实例对象调用。但类方法不能访问实例变量，只能访问类变量。类方法可以由类名直接调用，也可由实例对象进行调用。类方法中不能使用this或super关键字。

7.内部类与静态内部类的区别？

静态内部类相对与外部类是独立存在的，在静态内部类中无法直接访问外部类中变量、方法。如果要访问的话，必须要new一个外部类的对象，使用new出来的对象来访问。但是可以直接访问静态的变量、调用静态的方法；

普通内部类作为外部类一个成员而存在，在普通内部类中可以直接访问外部类属性，调用外部类的方法。

如果外部类要访问内部类的属性或者调用内部类的方法，必须要创建一个内部类的对象，使用该对象访问属性或者调用方法。

如果其他的类要访问普通内部类的属性或者调用普通内部类的方法，必须要在外部类中创建一个普通内部类的对象作为一个属性，外同类可以通过该属性调用普通内部类的方法或者访问普通内部类的属性

如果其他的类要访问静态内部类的属性或者调用静态内部类的方法，直接创建一个静态内部类对象即可。

8、static关键字的作用（5分）

答案：static可以用来修饰类的成员方法、类的成员变量，另外可以编写static代码块来优化程序性能。

1）static方法

static方法一般称作静态方法，由于静态方法不依赖于任何对象就可以进行访问，因此对于静态方法来说，是没有this的，因为它不依附于任何对象，既然都没有对象，就谈不上this了。并且由于这个特性，在静态方法中不能访问类的非静态成员变量和非静态成员方法，因为非静态成员方法/变量都是必须依赖具体的对象才能够被调用。

2）static变量

　　static变量也称作静态变量，静态变量和非静态变量的区别是：静态变量被所有的对象所共享，在内存中只有一个副本，它当且仅当在类初次加载时会被初始化。而非静态变量是对象所拥有的，在创建对象的时候被初始化，存在多个副本，各个对象拥有的副本互不影响。

　　static成员变量的初始化顺序按照定义的顺序进行初始化。

3）static代码块

static关键字还有一个比较关键的作用就是 用来形成静态代码块以优化程序性能。static块可以置于类中的任何地方，类中可以有多个static块。在类初次被加载的时候，会按照static块的顺序来执行每个static块，并且只会执行一次。

9、static关键字能修饰什么（5分）

答案：static关键字是Java中常用的关键字之一，可能最常用的就是修饰变量和方法了，但是仅仅知道这些还远远不够。问题虽然很小，但是却反映了对Java的了解程度。

static修饰变量和方法

static可以修饰变量，这个变量属于类本身，不需要创建实例就可以直接获取到值。

static可以修饰方法，这个方法属于类本身，同样，不要创建实例就可以通过类调用。

需要了解的是，static修饰的变量或方法属于类的静态资源，是所有实例共享的，另外静态方法内部是不能访问非静态方法的，因为静态资源是在类加载的时候就建立好的，而类加载时非静态方法需要类new的时候才能创建，一前一后的顺序所以导致了Java静态方法不能访问非静态资源的结果，当然反之肯定是可以的了。

static修饰代码块

static修饰的代码块是静态代码块，也具有静态的特点，属于类本身，在加载时只需要加载一次，也就是说，如果加载过这个代码块，就不会再加载了。

static修饰类

static修饰类只有一种情况，那就是这个类属于静态内部类，接触过Android开发的话可能遇见过很多这样的静态内部类，如WindowManager.LayoutParams类，LayoutParams就是WindowManager类下的静态内部类，它的源码如下所示：

public interface WindowManager extends ViewManager {

public static class LayoutParams extends ViewGroup.LayoutParams implements Parcelable {

//内部实现

}

}

10、使用static修饰属性或方法后，属性和方法有什么特征？（5分）

答案：static修饰属性或方法后，属性和方法不在属于某个特定的对象，而是所有共享，也可以说是static成员不依赖某个对象，在类加载时就被初始化。Static修饰的属性或方法，可以直接使用类名调用，而不用先实例化对象再调用。

* 11、请简述static静态修饰词可以修饰哪几种java组成元素（5分）

答案： 静态变量、静态方法、静态块、内部静态类和静态接口方法

12、静态内部类和非静态内部类的区别？

答案：

一 . 静态内部类可以有静态成员，而非静态内部类则不能有静态成员。   
二 . 静态内部类的非静态成员可以访问外部类的静态变量，而不可访问外部类的非静态变量；

三 . 非静态内部类的非静态成员可以访问外部类的非静态变量。

13、构造代码块、局部代码快、静态代码块他们执行的先后顺序？

答案：

先静态代码块

如果有对象，构造代码块执行

局部代码块按顺序执行

14、静态变量和非静态变量的区别是：

答案：

静态变量被所有的对象所共享，在内存中只有一个副本，它当且仅当在类初次加载时会被初始化。而非静态变量是对象所拥有的，在创建对象的时候被初始化，存在多个副本，各个对象拥有的副本互不影响。

static成员变量的初始化顺序按照定义的顺序进行初始化。

15、Static修饰什么？

答案：

静态变量、静态方法、静态代码块

六、程序实现题

1、创建一个包含嵌套类的类。在**main()**中创建其内部类的实例

|  |
| --- |
| **public** **class** One {  **static** **class** Two{}  **public** **static** **void** main(String[] args) {  // **TODO** Auto-generated method stub  One.Two two=**new** One.Two();  }  } |

2、编写程序，实现static修饰方法（10分）

|  |
| --- |
| public class Post {    private String title;  private String content;    private static int count = 0;    public Post(){  count++;  }    public static int getCount(){  return count;  }  } |

## 3、定义类MySystem，它有三个静态方法 ，请实现以下功能

（1） 静态方法int readInt()从控制台读入一个int型数。  
（2） 静态方法double readDouble()从控制台读入一个double型数。  
（3） 静态方法String readString()从控制台读入一个String。  
（4） 写main方法，测试这三个静态方法。

public class TestStatic {  
public static void main(String[] args) {  
 System.out.println("请输入整数回车确认：");  
 System.out.println("调用静态方法readInt返回输入的整数：" + readInt());  
 System.out.println("请输入浮点数回车确认：");  
 System.out.println("调用静态方法readDouble返回输入的整数："+readDouble());  
 System.out.println("请输入字符串回车确认：");  
 System.out.println("调用静态方法readString返回输入的整数：" + readString());  
  
}  
public static int readInt() {  
 Scanner sInt = new Scanner(System.in);  
 int i = sInt.nextInt();  
 return i;  
}  
public static double readDouble() {  
 Scanner sDouble = new Scanner(System.in);  
 double dou = sDouble.nextDouble();  
 eturn dou;  
}  
public static String readString() {   
 Scanner sString = new Scanner(System.in);  
 String str = sString.next();  
 return str;  
}  
}

4、分析以下程序运行的结果。

public class TestStaticClass {

public TestStaticClass(){

System.out.println("外部类的构造方法");

}

static {

System.out.println("外部类的静态代码块");

}

{

System.out.println("外部类的普通代码块");

}

private void outMethon(){

System.out.println("外部类的普通方法，这时候还没看到内部类有任何输出");

InnerCommonClass innerCommonClass=new InnerCommonClass();

innerCommonClass.innerMethord();

InnerStaticClass innerStaticClass=new InnerStaticClass();

innerStaticClass.innerMethord();

}

public class InnerCommonClass{

int a=10;

{

System.out.println("普通内部类的普通代码块----a="+a);

}

InnerCommonClass(){

System.out.println("普通内部类的构造方法----a="+a);

a=0;

}

public void innerMethord(){

System.out.println("普通内部类-----a="+a);

}

}

public static class InnerStaticClass{

int a=10;

static {

System.out.println("静态内部类的静态代码块");

}

InnerStaticClass(){

System.out.println("静态内部类的构造方法a="+a);

}

public static void innerStaticMethord(){

System.out.println("静态内部类的静态方法");

}

public void innerMethord(){

System.out.println("静态内部类的普通方法a="+a);

}

}

public static void main(String[] str){

System.out.println("主函数执行 new TestStaticClass()");

TestStaticClass test=new TestStaticClass();

test.outMethon();

}

}

外部类的静态代码块

主函数执行 new TestStaticClass()

外部类的普通代码块

外部类的构造方法

外部类的普通方法，这时候还没看到内部类有任何输出

普通内部类的普通代码块----a=10

普通内部类的构造方法----a=10

普通内部类-----a=0

静态内部类的静态代码块

静态内部类的构造方法a=10

静态内部类的普通方法a=10

5、 请举例说明静态内部类与非静态内部类的区别。

public class OutClassTest {

int out1=1;

static int out2=1;

void out(){

System.out.println("非静态");

}

static void outstatic(){

System.out.println("静态");

}

public class InnerClass{

void InnerClass(){

System.out.println("InnerClass!");

System.out.println(out1);

System.out.println(out2);

out();

outstatic();//静态内部类只能够访问外部类的静态成员

}

// static void inner(){} static int i=1; 非静态内部类不能有静态成员（方法、属性）

}

public static class InnerstaticClass{

void InnerstaticClass(){

System.out.println("InnerstaticClass");

// System.out.println(out1);out(); 静态内部类只能够访问外部类的静态成员

System.out.println(out2);

outstatic();

}

static void innerstatic(){} static int i=1;//静态内部类能有静态成员（方法、属性）

}

public static void main(String args[]){

OutClassTest a=new OutClassTest();

OutClassTest.InnerstaticClass b=new OutClassTest.InnerstaticClass();//创建静态内部类

OutClassTest.InnerClass c=a.new InnerClass();//创建非静态内部类

}

}

6、分析以下程序运行的结果。

public class HelloWorld {

private static int score = 84;

private static int a = 78;

private int b = 86;

public static class SInner {

int score = 91;

public void show() {

System.out.println("访问外部类中的score：" +HelloWorld.score);

System.out.println("访问内部类中的score：" + score);

System.out.println("访问外部类中的a：" + a);

System.out.println("访问外部类中的b：" + new HelloWorld().b);

}

}

public static void main(String[] args) {

SInner si = new SInner();

si.show();

}

}

访问外部类中的score：84  
访问内部类中的score：91  
访问外部类中的a：78  
访问外部类中的b：86

7、利用递归方法求5!。。

递归公式：f(n)=n\*f(n-1)

（10分）

|  |
| --- |
| public class Programme22 {  public static void main(String[] args) {  System.out.println(Fact(5));//输出结果值  }  //递归方法求阶乘的具体代码  private static int Fact(inti) {  if (i==1) {//1的阶乘为1  return 1;  }  returni\*Fact(i-1);//不断回调  }  } |

8、定义常量： 定义一个静态类 并在其他类中调用静态类的静态方法（10分）

|  |
| --- |
| public class Test10 {  public static void main(String[] args) {  // TODO Auto-generated method stub  Test10 test=new Test10();  Test1.play();  }  static class Test1{  public static void play(){  System.out.println("被调用了！");  }  }  } |

9、编写一个静态内部类并实例化其对象

public class Outer {

static class Inner{

}

public static void main(String[] args) {

Outer.Inner inner = new Outer.Inner();

}

}

10、创建一个名称为StaticDemo的类 并声明一个静态变量和一个普通变量。对变量分别赋予10和5的初始值。在main()方法中输出变量值。

（10分）

|  |
| --- |
| public class StaticDemo  {  private static int i=10;  private int y=5;  public static void main(String[] das)  {  StaticDemo s=new StaticDemo();  System.out.println("The static digit is "+car.i+"\n and the normal digit is "+s.y);  }  } |