Day09

**Key Point**

接口

内部类

方法内部类

成员内部类

静态内部类

匿名内部类

包

垃圾分代回收机制

# 一、 选择题

1. 下列各项说法中，关于内部类的说法错误的是：E.
   1. 方法内部类可以使用当前外部类中的成员变量
   2. 成员内部类可以使用当前外部类中的成员方法
   3. 静态内部类可以使用当前外部类中的静态方法
   4. 成员内部类中可以定义静态常量
   5. java.lang.String可以声明一个匿名内部类
2. 下列关于包的说法正确的是：CD
   1. 一个Java文件在中可以写多个package个语句
   2. 可以通过import来声明一个包
   3. 如果是同包类可以不用导包直接使用
   4. 对于java.lang包下的类，不需要手动导包即可使用
   5. import java.util.\*表示导入util包下的所有类以及子包下的类
3. 关于接口和抽象类，下列说法正确的是：AE
   1. 抽象类可以有构造方法，接口没有构造方法
   2. 抽象类可以有属性，接口没有属性
   3. 抽象类可以有非抽象方法，接口中都是抽象方法
   4. 抽象类和接口都不能创建对象
   5. 一个类最多可以继承一个抽象类，但是可以实现多个接口
4. Java 中的内部类包括ABCD
   1. 成员内部类
   2. 静态内部类
   3. 局部内部类
   4. 匿名内部类
   5. 公开内部类
   6. 抽象内部类
5. 有下面代码

**class OuterClass**{

**private int** value1 = 100; **private static int** value2 = 200; **public void** method(**int** value3){

**final int** value4 = 400;

**class InnerClass**{

**public void** print(){

//1

}

}

}

}

问：下面哪些代码放在//1 处能够编译通过？ACD

* 1. System.out.println(value1);
  2. System.out.println(value2);
  3. System.out.println(value3);
  4. System.out.println(value4);

# 二、 简答题

1. 请简述接口和抽象类的区别
2. 请简述垃圾分代回收机制
3. 代码改错：Myclass的m1方法权限修饰符为public

**interface IA**{

**void** m1();

**int** a = 100;

}

**class MyClass implements** IA{

**void** m1(){}

}

**public class TestInterface**{

**public static void** main(String args[]){ IA ia = **new** MyClass();

ia.m1(); System.out.println(IA.a);

}

}

1. 有如下代码：

**interface IA**{

**void** ma();

}

**interface IB extends** IA{

**void** mb();

}

**interface IC**{

**void** mc();

}

**interface ID extends** IB, IC{

**void** md();

}

* 1. 如果有一个类 ClassE 实现 ID 接口，如果不希望 ClassE 是抽象的，则需要实现哪些方法？mambmcmd
  2. 把下面的代码补充完整

**public class TestClassE**{

**public static void** main(String args[]){ IC ic = **new** ClassE();

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ //调用 ma 方法

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ //调用 mb 方法

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ //调用 mc 方法

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ //调用 md 方法

Test t=(Test) (ic);

t.ma();//调用 ma 方法

t.mb();//调用 mb 方法

ic.mc();//调用 mc 方法

t.md();//调用 md 方法

}

}

* 1. 写出下面代码的输出结果 t t t t t

**public class TestClassE**{

**public static void** main(String args[]){ IC ic = **new** ClassE(); System.out.println(ic **instanceof** IA); System.out.println(ic **instanceof** IB); System.out.println(ic **instanceof** IC); System.out.println(ic **instanceof** ID);

System.out.println(ic **instanceof** ClassE);

}

}

1. 写出下面代码的运行结果

**interface IA**{

**void** ma(IB ib);

}

**interface IB**{

**void** mb();

}

**class IAImpl implements** IA{

**public void** ma(IB ib){ System.out.println(“ma in IAImpl” ); ib.mb();

}

}

**class IBImpl implements** IB{

**private** IA ia;

**public void** setIa(IA ia){

**this**.ia = ia;

}

**public void** mb(){

System.out.println(“mb in IBImpl” );

}

**public void** method(){ ia.ma(**this**);

}

}

**public class TestMain**{

**public static void** main(String args[]){ IA ia = **new** IAImpl();

IBImpl ib = **new** IBImpl(); ib.setIa(ia);

ib.method();

}

}

6.写出下面代码执行的结果

in method

ma in anonymous inner class

in method

ma in local inner class

**interface IA**{

**void** ma();

}

**class MyClass** {

**public static void** method(IA ia){ System.out.println(“in method” ); ia.ma();

}

}

**public class TestInnerClass**{

**public static void** main(String args[]){ MyClass.method(**new** IA(){

**public void** ma(){

System.out.println(“ma in anonymous inner class” );

}

});

**class MyMaClass implements** IA{

**public void** ma(){

System.out.println(“ma in local inner class” );

}

}

MyClass.method(**new** MyMaClass());

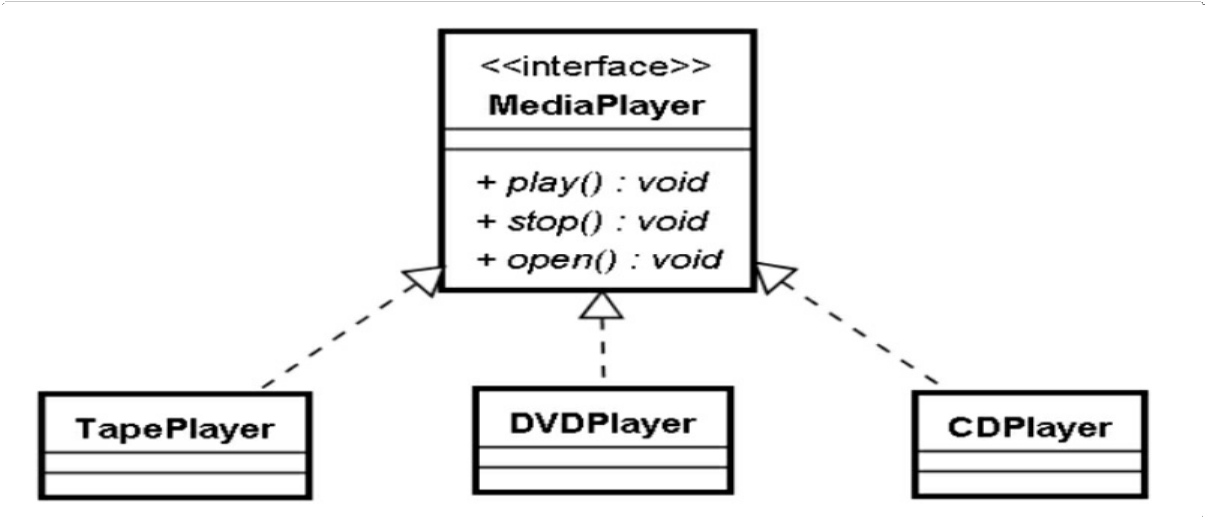
}

}

三、 编程题

1. 定义一个接口 MediaPlayer，表示家庭影院的一个设备。 MediaPlayer 中包含 play()， stop()， open()三个方法，分别表示播放、停止和开仓功能。MediaPlayer 有三个实现类，分别为： DVDPlayer，表示 DVD 播放器； CDPlayer，表示 CD 播放器；

TapePlayer，表示录音机（播放磁带）。类图如下：



创建一个遥控器 Controller 类。该遥控器有三个控制通道，可以分别控制三个设备。部分代码如下：

**class Controller**{

**private** MediaPlayer[] players;

**public** Controller(){

//构造函数中初始化 players 数组

}

//对相应的设备调用 play 方法**public void** play(**int** i){ players[i].play();

}

}

要求：

1. 完成 MediaPlayer 接口及其子类的代码。
2. 把 Controller 补充完整，完善其构造函数，并为其增加 stop(int i)和open(int i)方法