1、下列代码的运行结果为：

public class Animal {

public void show() {

System.out.println("我是动物");

}

}

public class Tiger extends Animal {

public void show() {

System.out.println("我是老虎");

}

}

public class Dog extends Animal {

public void show() {

System.out.println("我是哈士奇");

}

}

public class Test {

public static void main(String[] args) {

Animal one = new Animal();

one.show();

Animal two = new Tiger();

two.show();

Animal three = new Dog();

three.show();

}

}

1. 创建一个父类Animal，一个子类Cat，Animal three = new Cat();不是  
   A. 向上转型   B. 自动转型   C. 向下转型   D. 隐式转型
2. 下列代码怎么修改可以使其成功运行：  
   A. 删除掉标注3位置的one.fly( )  
   B. 标注1的Animal one=new Bird()修改为Animal one=new Animal()  
   C. 删除掉标注2位置的one.eat( )  
   D. 标注1的Animal one=new Bird()修改为Bird one=new Animal()

public class Animal {

public void eat() {

System.out.println("动物吃~~~");

}

}

public class Bird extends Animal {

public void eat() {

System.out.println("鸟吃虫子");

}

public void fly() {

System.out.println("鸟儿正在飞");

}

}

public class Test {

public static void main(String[] args) {

Animal one = new Bird(); //1

one.eat(); //2

one.fly(); //3

}

}

1. 下列关于instanceof说法不正确的是  
   A. instanceof 的返回值为true和false  
   B. instanceof可以用来判断对象是否可满足某个特定类型  
   C. 可以通过“A instanceof B"表示 A 类可以转型为B类  
   D. instanceof可放在if语句的条件表达式中
2. 已知父类Person,子类Man。判断类Person的对象person1是否满足类Man的实例特征，正确的语 句为  
         Person person1 = new Man();  
         Man man1 = new Man();  
   A. if (person1 instanceof Man)  
   B. if (man1 instanceof Person)  
   C. if (Person instanceof man1)  
   D. if (Man instanceof person1)
3. 在Java中，多态的实现不仅能减少编码的工作量，还能大大提高程序的可维护性及可扩展性，那 么实现多态的步骤包括以下几个方面除了  
   A. 子类重写父类的方法  
   B. 子类方法设置为final类型  
   C. 定义方法时，把父类类型作为参数类型；调用方法时，把父类或子类的对象作为参数传 D. 运行时，根据实际创建的对象类型动态决定使用哪个方法
4. 下面代码运行测试后，出现的结果是  
   A. 编译错误，错误位置在第一行  
   B. 编译错误，错误位置在第二行  
   C. 第一行和第二行都运行成功，输出结果为 儿子 女儿  
   D. 编译成功，但运行报错，错误位置在第二行

public class Father {

public void show() {

System.out.println("父亲");

}

}

public class Son extends Father {

public void show() {

System.out.println("儿子");

}

}

public class Daughter extends Father {

public void show() {

System.out.println("女儿");

}

}

public class Test {

public static void main(String[] args) {

Father s = new Son();

Father d = new Daughter();

Son s1 = (Son)s; //第一行

s1.show();

Daughter d1 = (Daughter)s; //第二行

d1.show();

}

}

1. 下面代码怎么修改可以编译时不报错（多选）  
   A. 在位置一处将SpeedBike类设为抽象类，同时将位置2处的speedup也设为抽象方法  
   B. 将位置一中的public改为final  
   C. 将位置二中speedup()方法改为抽象方法  
   D. 将位置二中speedup()方法中加入方法的实现

public abstract class Bike {

int colornum;

int brand;

int speed;

public abstract void speedup();

}

public class SpeedBike extends Bike { //位置一

public void speedup(); //位置二

}

1. 下列选项中，关于Java的抽象类和抽象方法说法不正确的是  
   A. 抽象类和抽象方法都通过abstract关键字来修饰  
   B. 抽象类中必须包含抽象方法  
   C. 抽象方法只有方法声明，没有方法实现  
   D. 子类如果不重写父类所有的抽象方法，则必须设置为抽象类

编程题：

应用继承和多态的思想，编写动物类，成员方法是动物叫声。写三个具体的类（猫、狗、羊）， 它们都是动物类的子类，并重写父类的成员方法。编写测试类，随机产生三种具体动物，调用叫 声这个方法。

1. 定义一个父类Animal类 属性：kind(种类) 方法：创建带参（kind为参数）构造方法 创建cry():void方法
2. 编写三个具体的子类Cat类、Dog类、Sheep类 分别重写父类中的 cry() 方法，输出信息分别为
   1. Cat类：小猫的叫声：喵喵喵~~~
   2. Dog类：小狗的叫声：汪汪汪~~~
   3. Sheep类：小羊的叫声：咩咩咩~~~
3. 编写测试类，首先生成长度为5的父类对象数组，然后通过循环依次向数组中存入数据，现 设定存储规则为：
   1. a) 每次随机产生一个0~2的正整数
   2. b) 若数值为 0，则生成一个 Cat 类的对象，存入数组
   3. c) 若数值为 1，则生成一个 Dog 类的对象，存入数组
   4. d) 若数值为 2，则生成一个 Sheep 类的对象，存入数组 最后循环输出数组成员，并分别调用 cry() 方法。

定义一个类People  
定义两个类Man 和Woman  
man有个属性是老婆,有一个方法是lol  
woman有个属性是老公,有一个方法是shopping,还有一个方法是生孩子  
50%概率是男孩儿,50%概率是女孩儿  
生孩子(先判断是否有老公,如果有就创建一个对象,有返回值)  
男孩儿就调用lol  
女孩就调用shopping