1．下列程序运行结果是（    ）。

private class Base {

    Base() {

        int i = 100;

        System.out.println(i);

    }

}

public class Pri extends Base {

    static int i = 200;

    public static voidmain(String argv[]) {

        Pri p = new Pri();

        System.out.println(i);

    }

}

A．编译错误                     B．200

C．100 200                     D．100

2．下列程序运行结果是（    ）。

class Base {

    Base() {

        int i = 100;

        System.out.println(i);

    }

}

public class Pri extendsBase {

    static int i = 200;

    public static void main(String argv[]) {

        Pri p = newPri();

        System.out.println(i);

    }

}

A．编译错误                     B．200

C．100  200                     D．100

3．在Java中，下面描述正确的是（    ）。

A．一个子类可以有多个父类，一个父类也可以有多个子类

B．一个子类可以有多个父类，但一个父类只可以有一个子类

C．一个子类只可以有一个父类，但一个父类可以有多个子类

D．上述说法都不对

4．设有文件Derived.java中代码如下：

public class Base extendsObject {

    String objType;

    public Base() {

        objType = "I am a Base type";

    }

}

public class Derived extendsBase {

    public Derived() {

        objType = "I am a Derived type";

    }

    public static void main(String args[]) {

        Derived d = newDerived();

    }

}

编译程序将出现何问题？（    ）

A．将创建 Base.class 和 Derived.class 两个文件

B．编译程序将指示第1行有问题

C．编译程序将在第7行出错

D．以上都不对

5．看下列程序

package a;

class Parent {

    private int i = 20;

    protected int j = 30;

    public int k = 40;

    int h = 50;

}

class Child extends Parent {

    void f() {

    }

}

在子类child的方法f()中不可以操作的变量是（    ）。

A．i                            B．j

C．k                            D．h

6．看下列程序

package a;

class Parent {

    private int i = 20;

    protected int j = 30;

    public int k = 40;

    int h = 50;

}

class Child extends Parent {

}

class GrandSon extends Child {

    void f() {

    }

}

在子类GrandSon的方法f()中不可以操作的变量是（    ）。

A．i                            B．j

C．k                            D．h

7．类的声明如下：

class A {

}

则类A的父类是（    ）。

A．没有父类                    B．本身

C．java.lang.Object            D．以上都不对

8．下列程序的运行结果是（    ）。

class Parent {

    int i = 20;

    int j = 30;

    void f() {

        System.out.print("" + i);

    }

}

class Child extends Parent {

    int i = 30;

    intk = 40;

    void f() {

        System.out.print(" " + i);

    }

    Void g() {

        System.out.print("" + k);

    }

    public static void main(String args[]) {

        Parent x = newChild();

        System.out.print(x.i);

        x.f();

        Child x1 = (Child) x;

        System.out.print("" + x1.i);

        x1.f();

    }

}

A．30 30 3030                  B．20 20 20 20

C．20 30 3030                  D．都不对

9．关于重载和重写的叙述正确的是（    ）。

A．重载是多态的一种，而重写不是

B．重载是子类中定义的方法和父类中某个方法相同

C．重写是一个类中多个同名的方法，并且方法的参数不同

D．重写方法时不允许降低方法的访问权限

10．哪个选项可以作为以下方法的覆盖方法？（    ）

public void add(int a) {…}

A．public void add(int b) {…}

B．void add(int a) {…}

C．public int add(int a) {…}

D．public void add(float a) {…}

11．（    ）是在子类中创建一个和父类具有一样特征的方法，特征包括方法名字、参数个数、参数类型和方法返回值类型。

A．覆盖                         B．重载

C．继承                        D．以上都不正确

12．第10行将调用的会是哪个方法？（    ）

1.class Person {

2.    public voidprintValue(int i, int j) { }

3.    public voidprintValue(int i) { }

4.}

5.public class Teacher extendsPerson {

6.    public voidprintValue() { }

7.    public voidprintValue(int i) { }

8.    public static void main(String args[]) {

9.        Person t = new Teacher();

10.       t.printValue(10);

11.   }

12.}

A．on line 2                    B．on line 3

C．on line 6                    D．on line 7

13．哪种访问组合可放在第2行add前和第6行的add前？（    ）

1.class SuperDuper {

2.    void add() {

3.    }

4.}

5.class Sub extends SuperDuper {

6.    void add() {

7.    }

8.}

A．line 2: public; line 6: private

B．line 2: protected; line 6: private

C．line 2: private; line 6: protected

D．line 2: public; line 6: protected

14．下列程序的运行结果是（    ）。

public class Test {

    public static void test() {

        this.print();

    }

    public static void print() {

        System.out.println("Test");

    }

    public static void main(String args[]) {

        test();

    }

}

A．输出Test

B．无输出结果

C．类编译错误，指示不能在static上下文中使用this

D．以上都不对

15．试完成下述程序片段（    ）。

public class Point {

int x, y;

public Point(intx, int y) {

(\_\_\_\_\_\_\_\_) = x;

(\_\_\_\_\_\_\_\_) = y;

   }

}

A．Point.x   Point.y            B．无解

C．x1  y1                       D．this.x  this.y

16．在子类构造方法的哪个地方可以调用超类的构造方法？（    ）

A．任何地方                     B．构造方法的第一条语句

C．构造方法的最后一条语句       D．不能在子类构造方法中调用超类的构造方法

17．哪个关键词在子类中用来访问与父类中一样的方法？（    ）

A．super                        B．this

C．static                       D．以上均不对

18．哪个关键词用来引用当前类的对象？（    ）

A．super                        B．this

C．static                       D．以上均不对

19．设有如下代码：

1.class Example {

2.    Stringstr;

3.    Example(){

4.       str= "example";

5.    }

6.    Example(Strings) {

7.       str= s;

8.    }

9. }

10.class Demo extends Example{

11.}

12.public class Test {

13.    public void f() {

14.        Exampleex = new Example("Good");

15.        Demod = new Demo("Good");

16.    }

17. }

以下哪行将导致错误？（    ）

A．第3行                       B．第6行

C．第10行                     D．第15行

20．下列程序的运行结果是（    ）。

class Parent {

    void test() {

        System.out.print("parent");

    }

}

public class Child extendsParent {

    void test() {

        super.test();

        System.out.print("child");

    }

    public static void main(String args[]) {

        Child x = newChild();

        x.test();

    }

}

A．parent child                 B．child

C．parent                       D．child parent

21．下列程序的运行结果是（    ）。

class Parent {

    Parent(String s) {

        s = "parent";

    }

    void test() {

        System.out.print("parent");

    }

}

public class Child extendsParent {

    void test() {

        super.test();

        System.out.print("child");

    }

    public static void main(String args[]) {

        Child x = newChild();

        x.test();

    }

}

A．parent child                 B．child

C．parent                       D．编译错误

22．下列程序的运行结果是（    ）。

class Parent {

    Parent(String s) {

        s = "parent";

    }

    void test() {

        System.out.print("parent");

    }

}

public class Child extendsParent {

    Child(String s) {

        s = "child";

    }

    void test() {

        super.test();

        System.out.print("child");

    }

    public static void main(String args[]) {

        Child x = newChild();

        x.test();

    }

}

A．parent child                 B．child

C．parent                       D．编译错误

23．以下代码运行结果是（    ）。

class Base {

}

class Sub extends Base {

}

class Sub2 extends Base {

}

class Cex {

    public static void main(String argv[]) {

        Base b = newBase();

        Sub s = (Sub) b;

    }

}

A．语法错误                    B．编译错误

C．运行异常                     D．以上都不对

24．Sub类的main方法执行结果为（    ）。

classBase {

    public void show(int i) {

        System.out.print(" Value is "+ i);

    }

}

classSub extends Base {

    public void show(int j) {

        System.out.print(" It is " +j);

    }

    public void show(String s) {

        System.out.print(" I was passed" + s);

    }

    public static void main(String args[]) {

        Base bl = new Base();

        Base b2 = new Sub();

        bl.show(5);

        b2.show(6);

    }

}  
A．It is 6 Value is 5

B．This value is 5 It is 6

C．Value is 5 It is 6

D．This value It is 6

25．如果一个成员变量声明时必须赋给初值，而且不能再发生变化，那么这个成员变量是（    ）。

A．私有变量                     B．最终变量（常量）

C．受保护的变量                 D．都不对

26．如何定义一个不能有子类的类Key？（    ）

A．class Key {}                B．final class Key { }

C．public classKey { }         D．class Key { finalint i; }

27．什么样的方法不能被重写？（    ）

A．私有方法                     B．最终方法（final方法）

C．受保护的方法                 D．都不对

28．在Java中，关于final关键词的说法正确的是（    ）。

A．如果修饰变量，就等同一个常量，定义时可以不赋值，一旦赋值不能再修改

B．如果修饰类，则该类只能被一个子类继承

C．如果修饰方法，则该方法不能在子类中被覆盖

D．如果修饰方法，则该方法所在的类不能被继承

29．下列有关整型的最终属性i的定义正确的是（    ）。

A．static int i;

B．final i;

C．static finalint i = 50;

D．final float i= 1.2;

30．可以限制一个方法重载的声明语句是（    ）。

A. final void test(){ }

B. final test() { }

C. static void test(){ }

D. abstract finalvoid test() { }

**二、简答题**

简述Overload和Override的区别，并说明Overload的方法是否可以有不同的返回值类型？

**三、编程题**

1．写一个Java应用程序，主要是体现父类子类间的继承关系。父类：鸟，子类：麻雀、鸵鸟、鹰。子类继承父类的一些特点，如都是鸟的话就都会有翅膀、两条腿等，但它们各自又有各自的特点，如麻雀的年龄、体重；鸵鸟的身高、奔跑速度；鹰的捕食、飞翔高度等。

2．实现一个名为Person的类和它的子类Employee，Employee有两个子类Faculty和Staff。具体要求如下：

（1）Person类中的属性有：姓名name（String类型）、地址address（String类型）、电话号码telephone（String类型）和电子邮件地址email（String类型）。

（2）Employee类中的属性有：办公室office（String类型）、工资wage（double类型）和受雇日期hiredate（String类型）。

（3）Faculty类中的属性有：学位degree（String类型）、级别level（String类

型）。

（4）Staff类中的属性有：职务称号duty（String类型）。

3．在包a中编写一个类Father，具有属性：年龄（私有）、姓名（公有）；具有功能：工作（公有）、开车（公有）。在包a中编写一个子类Son，具有属性：年龄（受保护的）、姓名；具有功能：玩（私有）、学习（公有）。最后在包b中编写主类Test，在主类的main方法中测试类Father与类Son。

4．编写一个Car类，具有final类型的属性品牌，具有功能drive；定义其子类Aodi和Benchi，具有属性：价格、型号；具有功能：变速；定义主类E，在其main方法中分别创建Aodi和Benchi的对象，并测试对象的特性。

5．定义如下图4-1所示的三个类，People类中的三个方法分别输出一些信息，ChinesePeople和AmericanPeople类重写父类的三个方法。

图4-1 类关系图

6．按要求编写一个Java应用程序：

（1）定义一个矩形类Rect，包含两个protected属性：矩形的宽width和高height。一个带有两个参数的构造方法，用于将width和height属性初始化。一个不带参数的构造方法，将矩形的宽和高都初始化为10。还有两个一般方法：求矩形面积的方法getArea()和求矩形周长的方法getPerimeter()。

（2）通过继承Rect类编写一个具有确定位置的矩形类PlainRect，其确定位置用矩形的左上角坐标来标识，有两个属性：矩形左上角坐标startX和startY。两个构造方法：一个是带4个参数的构造方法，用于对startX、startY、width和height属性初始化。一个是不带参数的构造方法，将矩形初始化为左上角坐标、长和宽都为0的矩形。

添加一个方法：判断某个点是否在矩形内部的方法isInside(doublex,double y)。如在矩形内，返回true, 否则，返回false。提示：点在矩形内是指满足条件x >= startX && x <= (startX + width) && y >=(startY - height) && y <= startY

（3）编写PlainRect类的测试程序，创建一个左上角坐标为（10，10），长为20，宽为10的矩形对象；计算并打印输出矩形的面积和周长；判断点(25.5，13)是否在矩形内，并打印输出相关信息。

7．按要求编写一个Java应用程序：

（1）定义一个类，描述一个矩形，包含有长、宽两种属性和计算面积方法。

（2）编写一个类，继承自矩形类，同时该类描述长方体，具有长、宽、高属性和计算体积的方法。

（3）编写一个测试类，对以上两个类进行测试，创建一个长方体，定义其长、宽、高，输出其底面积和体积。

8．编写一个类A，该类创建的对象可以调用方法f输出小写的英文字母表。然后再编写一个A类的子类B，要求子类B必须继承A类的方法f（不允许重写），子类B创建的对象不仅可以调用方法f输出小写的英文字母表，而且可以调用子类新增的方法g输出大写的英文字母表。最后编写主类C，在主类的main方法中测试类A与类B。

9．编写一个Java应用程序，该程序包括3个类：Monkey、People类和主类E。要求：

（1）Monkey中有个构造方法：Monkey (String s)，并且有个public void speak()方法，在speak方法中输出“咿咿呀呀……”的信息。

（2）People是Monkey的子类，在People类中重写父类的方法speak(),在speak方法中输出“别说话！”的信息。

（3）在People类中新增方法void think()，在think方法中输出“认真思考！”。

（4）在主类E的main方法中创建Monkey与People的对象来测试这两个类的功能。

10．编写一个Java应用程序，设计一个汽车类Vehicle，包含的属性有车轮个数wheels和车重weight。小车类Car是Vehicle的子类，其中包含的属性有载人数loader。卡车类Truck是Car类的子类，其中包含的属性有载重量payload。每个类都有构造方法和输出相关数据的方法。最后，写一个测试类来测试这些类的功能。

11．编写一个Shape类，具有属性：周长和面积；定义其子类三角形和矩形，分别具有求周长的方法。定义主类E，在其main方法中创建三角形和矩形类的对象，并赋给Shape类的对象a、b，使用对象a、b来测试其特性。

12．编写一个Java应用程序，该程序包括3个类： A类、B类和主类E。其中类B是类A的子类，在子类B中新增了成员变量和成员方法，并且隐藏了父类A的成员变量和重写了父类A的成员方法。在主类E的main方法中，创建类B的对象并赋给父类A的对象a，使用上转型对象a来测试上转型对象的一些特性。

13．实现如图4-2所示的类之间的继承关系，并编写Music类来测试这些类。

图4-2 类之间的继承关系

4.5 精选习题参考答案

**一、单项选择题**

1．A         2．C        3．C        4．B        5．A        6．A

7．C        8．C        9．D        10．A       11．A       12．D

13．C       14．C        15．D       16．B       17．A       18．B

19．D       20．A        21．D       22．D       23．C       24．C

25．B       26．B        27．B       28．C       29．C       30．A

**二、简答题**

overload是指多个方法具有相同的名字，但这些方法的参数不同。方法的参数不同是指：参数的个数不同或者参数的类型不同，方法的返回值类型和参数的名字不参与比较，也就是说如果两个方法的名字相同，即使返回值类型相同，也必须保证参数不同。

override是父类与子类之间多态性的一种表现，子类override父类方法时，必须满足的条件是子类override的方法与父类的方法同名、返回值类型相同、参数个数和参数类型相同，而且子类方法的权限不低于父类方法的权限。

overload的方法可以有不同的返回值类型。

**三、编程题**

1．

// 父类

public class Bird {

    int leg = 2;

    int flychi = 2;

    public void sing() {

        System.out.println("I'm abird!");

    }

    public void fly() {

        System.out.println("I canfly!");

    }

    public void grow() {

        System.out.println("I have twowings and two legs!");

    }

}

// 麻雀

class Sparrowextends Bird {

    int age;

    int weight;

    public void sing() {

        System.out.println("I'm asparrow!");

    }

    public void printAge() {

        System.out.println("My age is" + age);

    }

    public void printWeight() {

        System.out.println("My weight is" + weight);

    }

    public void setAge(int age) {

        this.age = age;

    }

    public void setWeight(int wight) {

        this.weight = weight;

    }

}

// 鸵鸟

class Ostrichextends Bird {

    int speed;

    int height;

    public void sing() {

        System.out.println("I'm aostrich!");

    }

    public void fly() {

        System.out.println("I can'tfly!");

    }

    public void printSpeed() {

        System.out.println("My speed is" + speed);

    }

    public void printHeight() {

        System.out.println("My height is" + weight);

    }

    public void setSpeed(int speed) {

        this.speed = speed;

    }

    public void setHeight(int hight) {

        this.height = height;

    }

}

// 鹰类答案略。

2．

class Person {

    String name = "li";

    String address = "liaoning ";

    String telephone = "0411-86111136";

    String email = "happy@tom.com";

}

class Employeeextends Person {

    String office = "220";

    double wage = 6000;

    String hiredate = "20070101";

}

class Facultyextends Employee {

    String degree = "doctor";

    String level = "middle";

}

class Staff extendsEmployee {

    String duty = "profession";

}

3．

// 文件Father.java

package a;

public class Father{

    private int age = 42;

    public String name = "laozhang";

    public void work() {

        System.out.println("workhard");

    }

    public void drive() {

        System.out.println("I candrive.");

    }

}

// 文件Son.java

package a;

public class Son extendsFather {

    protected int age = 18;

    String name = "xiaozhang";

    private void play() {

        System.out.println("enjoy thelife.");

    }

    public void study() {

        System.out.println("ai!");

    }

}

// 文件Test.java

package b;

import a.\*;

public class Test {

    public static void main(String[] args) {

        Father f = new Father();

        f.drive();

        f.work();

        System.out.println(f.name);

        Son s = new Son();

        s.study();

    }

}

程序的输出结果：

I can drive.

work hard

laozhang

ai!

4．

class Car {

    final String mark = "X1";

    void drive() {

        System.out.println("平安驾驶");

    }

}

class Audi extendsCar {

    double price = 10;

    String series = "Q5";

    void changeSpeed() {

        System.out.println("低速慢行");

    }

}

class Benz extendsCar {

    double price = 200;

    String series = "V6";

    void changeSpeed() {

        System.out.println("礼让三先");

    }

}

class E {

    public static void main(String[] args) {

        Audi ad = new Audi();

        System.out.println("品牌：" +ad.mark);

        System.out.println("价格：" +ad.price);

        System.out.println("型号：" +ad.series);

        ad.drive();

        ad.changeSpeed();

        Benz bc = new Benz();

        System.out.println("品牌：" +bc.mark);

        System.out.println("价格：" +bc.price);

        System.out.println("型号：" +bc.series);

        bc.drive();

        bc.changeSpeed();

    }

}

程序的输出结果：

品牌：X1

价格：10.0

型号：Q5

平安驾驶

低速慢行

品牌：X1

价格：200.0

型号：V6

平安驾驶

礼让三先

5．

class People {

    protected double height = 1.8;

    protected double weight = 150;

    public void speakHello() {

        System.out.println("hello");

    }

    public void showAverageHeight() {

        System.out.println(1.72);

    }

    public void showAverageWeight() {

        System.out.println(120);

    }

}

class ChinesePeopleextends People {

    public void chineseKungfu() {

        System.out.println("坐如钟，站如松，睡如弓。");

    }

    public void speakHello() {

        System.out.println("你好");

    }

    public void getAverageHeight() {

        System.out.println(1.75);

    }

    public void getAverageWeight() {

        System.out.println(130);

    }

}

classAmericanPeople extends People {

    public void americanBoxing() {

        System.out.println("直拳，勾拳。");

    }

    public void speakHello() {

        System.out.println("hi");

    }

    public void showAverageHeight() {

        System.out.println(1.75);

    }

    public void showAverageWeight() {

        System.out.println(135);

    }

}

class TestPeople {

    public static void main(String[] args) {

        ChinesePeople chinese = newChinesePeople();

        chinese.speakHello();

        chinese.showAverageHeight();

        chinese.showAverageWeight();

        chinese.chineseKungfu();

        AmericanPeople american = newAmericanPeople();

        american.speakHello();

        american.showAverageHeight();

        american.showAverageWeight();

        american.americanBoxing();

    }

}

程序的输出结果：

你好

1.72

120

坐如钟，站如松，睡如弓。

hi

1.75

135

直拳，勾拳。

6．

class Rect {

    protected double width;

    protected double height;

    Rect(double width, double height) {

        this.width = width;

        this.height = height;

    }

    Rect() {

        width = 10;

        height = 10;

    }

    double getArea() {

        return width \* height;

    }

    double getPerimeter() {

        return (width + height) \* 2;

    }

}

class PlainRectextends Rect {

    double startX, startY;

    PlainRect(double startX, double startY,

        double width, double height) {

        super(width, height);

        this.startX = startX;

        this.startY = startY;

    }

    PlainRect() {

        super(0, 0);

        startX = 0;

        startY = 0;

    }

    boolean isInside(double x, double y) {

        boolean rtn1 = (x >= startX)&& (x <= (startX + width));

        boolean rtn2 = (y <= startY)&& (y >= (startY - height));

        return rtn1 && rtn2;

    }

}

class PlainRectTest{

    public static void main(String[] args) {

        PlainRect rc = new PlainRect(10, 10, 20,10);

        System.out.println(rc.getArea());

        System.out.println(rc.getPerimeter());

        boolean flg = rc.isInside(25.5, 13);

        if (flg) {

            System.out.println("(25.5，13)在矩形内。");

        } else {

            System.out.println("(25.5，13)不在矩形内。");

        }

    }

}

程序的输出结果：

200.0

60.0

(25.5，13)不在矩形内。

7.

class Rectangle {

    double length;

    double width;

    Rectangle(double length, double width) {

        this.length = length;

        this.width = width;

    }

    double getArea() {

        return length \* width;

    }

}

class Rectangularextends Rectangle {

    double height;

    Rectangular(double length, double width,double height) {

        super(length, width);

        this.height = height;

    }

    double getVolume() {

        return length \* width \* height;

    }

}

class RectangleTest{

    public static void main(String[] args) {

        Rectangular cft = new Rectangular(10,20, 5);

        System.out.println("长方体的底面积："+ cft.getArea());

        System.out.println("长方体的体积："+ cft.getVolume());

    }

}

程序的输出结果：

长方体的底面积：200.0

长方体的体积：1000.0

8.

class A {

    void f() {

        System.out.println("A输出小写的英文字母表：");

        for (char c = 'a'; c <= 'z'; c++) {

            System.out.print(" " + c);

        }

        System.out.println();

    }

}

class B extends A {

    void g() {

        System.out.println("B输出大写的英文字母表：");

        for (char c = 'A'; c <= 'Z'; c++)

            System.out.print(" " + c);

        System.out.println();

    }

}

class C {

    public static void main(String args[]) {

        A a = new A();

        B b = new B();

        a.f();

        b.f();

        b.g();

    }

}

程序的输出结果：

A输出小写的英文字母表：

a b c d e f g h i j k l m n o p q r s t u v w x y z

A输出小写的英文字母表:

a b c d e f g h i j k l m n o p q r s t u v w x y z

B输出大写的英文字母表:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

9．

class Monkey {

    Monkey(String s) {

    }

    public void speak() {

        System.out.println("咿咿呀呀……");

    }

}

class Peopleextends Monkey {

    People() {

        super("");

    }

    public void speak() {

        System.out.println("别说话！");

    }

    void think() {

        System.out.println("认真思考！");

    }

}

class E {

    public static void main(String[] args) {

        Monkey yuan = new Monkey("");

        yuan.speak();

        People p = new People();

        p.speak();

        p.think();

    }

}

程序的输出结果：

咿咿呀呀……

别说话！

认真思考！

10．

class Vehicle {

    int wheels;

    double weight;

    Vehicle(int w1, double w2) {

        wheels = w1;

        weight = w2;

    }

    void print1() {

        System.out.println("Vehicle 车轮：" +wheels +

            "; 车重：" +weight);

    }

}

class Car extendsVehicle {

    int loader;

    Car(int l, int w1, double w2) {

        super(w1, w2);

        loader = l;

    }

    void print2() {

        System.out.println("Car 车轮：" +wheels +

            "; 车重：" +weight + "; 人数：" + loader);

    }

}

class Truck extendsCar {

    double payload;

    Truck(double p, int l, int w1, double w2) {

        super(l, w1, w2);

        payload = p;

        loader = l;

    }

    void print3() {

        System.out.println("Truck 车轮：" +wheels + "; 车重：" +

            weight + "; 人数：" +loader + "; 有载重量:" + payload);    }

}

class Test {

    public static void main(String[] args) {

        Vehicle v = new Vehicle(4, 50);

        v.print1();

        Car c = new Car(4, 30, 4);

        c.print2();

        Truck t = new Truck(4, 100, 2, 80);

        t.print3();

    }

}

程序的输出结果：

Vehicle 车轮：4; 车重：50.0

Car 车轮：30; 车重：4.0; 人数：4

Truck 车轮：2; 车重：80.0; 人数：100; 有载重量:4.0

11．

class Instrument {

    public void play() {

        System.out.println("弹奏乐器");

    }

}

class Wind extendsInstrument {

    public void play() {

        System.out.println("弹奏Wind");

    }

    public void play2() {

        System.out.println("调用Wind的play2");

    }

}

class Brass extendsInstrument {

    public void play() {

        System.out.println("弹奏Brass");

    }

    public void play2() {

        System.out.println("调用Brass的play2");

    }

}

class Music {

    public static void tune(Instrument i) {

        i.play();

    }

    public static void main(String args[]) {

        Wind w = new Wind();

        tune(w);

        tune(new Brass());

    }

}

程序的输出结果：

弹奏Wind

弹奏Brass

12．

class Shape {

    double length, area;

    double getLength() {

        return length;

    }

}

class Triangleextends Shape {

    double a, b, c;

    Triangle(double a, double b, double c) {

        this.a = a;

        this.b = b;

        this.c = c;

    }

    double getLength() {

        return a + b + c;

    }

}

class Rctangleextends Shape {

    double width, length;

    Rctangle(double length, double width) {

        this.length = length;

        this.width = width;

    }

    double getLength() {

        length = (width + length) \* 2;

        return length;

    }

}

class E {

    public static void main(String[] args) {

        Triangle t = new Triangle(3, 4, 5);

        Rctangle j = new Rctangle(2, 3);

        Shape a = t;

        System.out.println(a.getLength());

        Shape b = j;

        System.out.println(b.getLength());

    }

}

程序的输出结果：

12.0

10.0

13．

class A {

    int i = 1;

    int j = 10;

    void printA() {

        System.out.println("printA ofA");

    }

    void printB() {

        System.out.println("printB ofA");

    }

}

class B extends A {

    int j = 20;

    int k = 200;

    void printB() {

        System.out.println("printB ofB");

    }

    void printC() {

        System.out.println("printC ofB");

    }

}

class E {

    public static void main(String[] args) {

        A a = new B();

        System.out.println("a.i = " +a.i); // extends

        System.out.println("a.j = " +a.j); // 隐藏

        a.printA(); // extends

        a.printB(); // 重写

    }

}

程序的输出结果：

a.i = 1

a.j = 10

printA of A

printB of B