

第四章作业

4-1. 修饰类的访问控制有哪些?修饰类中成员的访问控制符有哪些?各有什么含义作用?请举例说明.

修饰类的访问控制:public, protected;

修饰类中成员的访问控制符:private, package protected , public;

4-2. 构造方法特殊在哪里?构造方法什么时候执行?被谁调用?

1. 这能在创建对象用 new 命令调用;
2. 构造方法是在创建对象时被自动调用的;
3. 构造方法名必须与类名相同, 没有返回值, 可以有参量, 并且可以被重载;

4-3. 关键字 static 可以修饰类的那些组成部分?实例变量和类变量的区别是什么?

1. 类的属性和方法;
2. 实例变量是非 static 的成员变量, 而类变量是所有对象共有的变量;

4-4 静态属性有什么特点?类的对象可以访问或修改静态属性吗?

1. 是类的属性, 不属于某个对象;

2. 可以访问或修改;

4-5 构造方法重载的特点是什么?有什么作用?

1. 方法名不变, 形式参数会有区别;

2. 用于创建对象传递给对象的参数可以多样化

4-6. 什么是包?如何创建包?包物理上对应什么操作系统中的实体?

1. 包是相关一组类的集合;

2. package 语句;

3. 文件夹;

4-7. 试写出创建一个名为 MyPackage 包的语句, 这个语句应该放在程序的什么位置?

1. package MyPackage;

2. 第一行;

4-8. 试写出引用 MyPackage 包中所有类的语句, 而引用 MyPackage 包中的一个 MyClass1 的语句;

1. import MyPackage.*;

2. import Mypackage.MyClass1;

4-9 阅读下面的程序, 说明他们的输出.

```
class MyClass{

    int data;

    MyClass(int d)

    {

        data=d;

    }

    int getData()

    {

        return data;

    }

    void setData(int d)

    {

        data=d;

    }

}

public class Test9{

    public static void main(String[] args)

    {

        MyClass myobj, myref;

        myobj = new MyClass(-1);
```

```

        myref = myobj;

        System.out.println("the original data
is:"+myobj.getData());

        myref.setData(10);

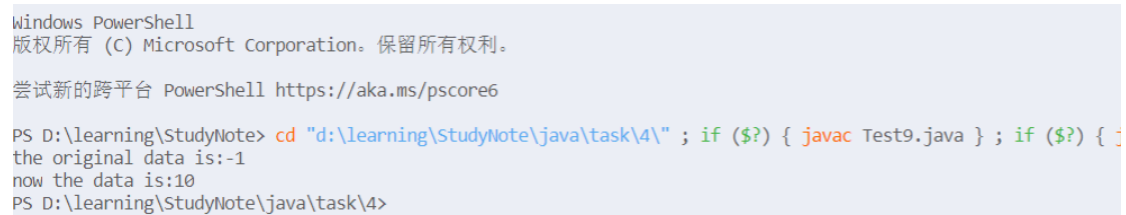
        System.out.println("now the data
is:"+myobj.getData());

    }

}

```

the original data is:-1 now the data is:10



```

Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/powershell

PS D:\learning\StudyNote> cd "d:\learning\StudyNote\java\task\4\" ; if ($?) { javac Test9.java } ; if ($?) { java Test9 }
the original data is:-1
now the data is:10
PS D:\learning\StudyNote\java\task\4>

```

4-10. 编写一个类实现秒表的功能. 要求实现开始计时, 停止计时, 分针秒针归零, 获得分针示数, 获得秒针示数, 获得总时间的毫秒数等功能.

```

public class Test10{

    int minute;//分针示数

    double seconds;//秒针示数

    void start();//开始计时

    void stop();//停止计时

```

```

void reZero(); //分针秒针归零

int getMinute(); //获得分针示数

double getSeconds(); //获得秒针示数

int getToatal(); //获得总时间的毫秒数
}

```

4-11. 编写一个类实现复数的运算, 要求至少实现复数相加, 复数相减, 复数相乘等功能;

```

import java.util.Scanner;

public class Test11{

    static void add(double a, double b, double c, double d)

    {

        System.out.println((a+c)+"i"+"(b+d)");

    }

    static void subtract(double a, double b, double c, double
d)

    {

        System.out.println((a-c)+"i"+"(b-d)");

    }

    static void mutiply(double a, double b, double c, double
d)

    {

```

```

        System.out.println((a*d+b*c)+"i"+"(b*d-a*c));
    }

    public static void main(String args[])
    {

        double a,b,c,d;

        Scanner input =new Scanner(System.in);

        a=input.nextDouble();

        b=input.nextDouble();

        c=input.nextDouble();

        d=input.nextDouble();

        add(a,b,c,d);

        subtract(a,b,c,d);

        mutiply(a,b,c,d);

    }

}

PS D:\learning\StudyNote\java\task\4> cd "d:\learning\StudyNote\java\task\4\" ; if ($?) { javac Test11.java } ; if ($?) { java Test11 }
1 2 3 4
4.0i+(6.0)
-2.0i+(-2.0)
10.0i+(5.0)
PS D:\learning\StudyNote\java\task\4>

```

4-12. 编程创建一个 Box 类. 要求:定义三个实例变量分别表示立方体的长宽高, 定义一个构造方法对这三个变量进行初始化, 然后定义一个方法求立方体的体积; 创建一个对象, 求给定尺寸的立方体的体积;

```

public class Test12{

    public static void main(String[] args)

```

```

    {
        Box a= new Box(1,2,3);

        System.out.println(a.getSize());
    }
}

class Box{
    double length,width,height;

    public Box(double l,double w,double h)
    {
        length = l;
        width = w;
        height = h;
    }

    public double getSize()
    {
        double s=length*(width*height);

        return s;
    }
}

```

```

PS D:\learning\StudyNote\java\task\4> cd "d:\learning\StudyNote\java\task\4\" ; if ($?) { javac Test12.java } ; if ($?) { java Test12 }
6.0
PS D:\learning\StudyNote\java\task\4>

```

4-13. 学生类的创建与使用.

1. 创建一个 student 类, 包括的域有学号, 班号, 姓名, 性别, 年龄等, 都是 private 类型;
2. 声明一个构造方法, 以初始化对象的所有的域;
3. 声明分别获得个属性的各个 Public 方法;
4. 声明修改个属性的各个 public 方法;
5. 声明一个 public 型的 toString() 方法, 把该类的所有域信息组合成一个字符串;
6. 在类中声明统计班级总人数的私有域 count 得到班级总人数的 public 方法(可在构造方法中进行 Student 对象个数的增加);
7. 将类 student 放在子包 Student 中;
8. 在子 包 student 外, 创建测试类 student 的主类;在主类中, 使用 student 类创建两个 student 对象, 输出对象的所有域信息;修改对象的姓名与年龄, 修改后显示各对象的姓名和年龄;你叫两个 student 对象的年龄大小, 输出年龄较大的 Student 对象;

```
import student.Student;
```

```
class Test13{
```

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

```
        Student a=new Student(1,3,"Lucre","boy",20);
```

```
        Student b=new Student(23,5,"Kiwis","boy",20);
```

```
        Student c=new Student(24,5,"XiaoWu","boy",20);
```



```
System.out.println(a.getId()+", " +a.getClassId()+', '+a.getName()+', '+a.getSex()+', '+a.getAge());
```

```
    a.setAge(19);
```

```
    a.setName("Panda");
```

```
    a.setSex("girl");
```

```
System.out.println(a.toString());
```

```
if(a.getAge()>b.getAge())
```

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

```
else System.out.println(b);
```

```
}
```

```
}
```

```
package student;
```

```
public class Student{
```

```
    private int id;
```

```
    private int classId;
```

```
    private String name;
```

```
    private String sex;
```

```
    private int age;
```

```
    private static int count=0;
```

```
    public Student(int id,int classId,String name,String sex,int age)
```

```
{

    this.id=id;

    this.classId=classId;

    this.name=name;

    this.sex=sex;

    this.age=age;

    count++;

}

public int getId()

{

    return id;

}

public int getClassId() {

    return classId;

}

public String getName()

{

    return name;

}

public String getSex()

{

    return sex;
```

```
}

public int getAge()

{

    return age;

}

public void setId(int x)

{

    this.id=x;

}

public void setClassId(int x)

{

    this.classId=x;

}

public void setName(String x)

{

    this.name=x;

}

public void setSex(String x)

{

    this.sex=x;

}

public void setAge(int x)
```

```

    {

        this.age=x;

    }

    public String toString()

    {

        return

        "id="+id+", classId="+classId+", name="+name+", sex="+sex+", ag

        e="+age;

    }

    public int getCount()

    {

        return count;

    }

}

```

1, 3, Lucre, boy, 20

id=1, classId=3, name=Panda, sex=girl, age=19

id=23, classId=5, name=Kiwis, sex=boy, age=20

```

PS D:\learning\StudyNote\java\task\4> cd "d:\learning\StudyNote\java\task\4\" ; if ($?) { javac Test13.java } ; if ($?) { java Test13 }
1,3,Lucre,boy,20
id=1,classId=3,name=Panda,sex=girl,age=19
id=23,classId=5,name=Kiwis,sex=boy,age=20

```

4-14 定义一个学生类和一个课程类, 要求能够表示现实生活中一个学生可以选择多门课程的情况. 编写程序进行测试;

```
class Test14{  
    public static void main(String args[])  
    {  
        Student a=new Student("Chengfen");  
        a.select("math");  
        a.select("English");  
    }  
}
```

```
public class Student{  
    Curriculum t[]=new Curriculum[20];  
    String name;  
    int num;  
    public Student(String s)  
    {  
        name=s;  
        num=0;  
    }  
    public void select(String s)  
    {  
        t[num++]=new Curriculum(s);  
    }  
}
```

```

}

public class Curriculum{

    String name;

    public Curriculum(String s)

    {

        name=s;

    }

}

```

4-15 定义一个方程类, 一求方程的两个实根; 请将测试主类和方程定义在不同的包中;

```

import equation.*;

class Test15{

    public static void main(String args[])

    {

        Equation a= new Equation(4,5,1);

        System.out.println("two results are:"+a.solve1()+"

"+a.solve2());

    }

}

package equation;

import java.lang.Math;

```

```

public class Equation{
    private int a,b,c;

    public Equation(int a,int b,int c)
    {
        this.a=a;
        this.b=b;
        this.c=c;
    }

    public double solve1()
    {
        return (-b+Math.sqrt(b*b-4*a*c))/2*a;
    }

    public double solve2()
    {
        return (-b-Math.sqrt(b*b-4*a*c))/2*a;
    }
}

```

```

PS D:\learning\StudyNote\java\task\4\task14> cd "d:\learning\StudyNote\java\task\4\" ; if ($?) { javac Test15.java } ; if ($?) { java Test15 }
two results are:-4.0 -16.0
PS D:\learning\StudyNote\java\task\4>

```

第五章

5-1 试描述继承下的父类和子类的概念;父类和子类有何关系?

- 1.父类是子类的一般化,而子类是父类的特殊化;
- 2.子类是父类的公共性基础上的延伸与扩展;

5-2 什么是单重继承?什么是多重继承?Java 采用什么继承

- 1.一个父类只有一个子类是单继承;
- 2.一个父类可以有多个子类是多继承;
- 3.多继承;

5-3 Java 中如何定义继承关系?写出定义类库中的类 `java.awt.Frame` 的子类 `MyFrame` 的类头的语句.

- 1.`extend` 是语句
- 2.`import java.awt.Frame;`
`class MyFrame extends Frame`

5-4 关键字 `protected` 的作用是什么?什么情况下比较好?

- 1.父类定义的 `protected` 成员在子类和同一包内可以直接使用;
- 2.如果确实需要在子类中修改超类的成员;

5-5 什么叫方法的重新定义?

在子类中重新定义父类的同名方法;如果子类重新定义了同名方法,我们则可以:
`super.function();`调用父类的同名方法;

5-6 试解释构造方法重载作用.一个构造方法如何调用同类的其他方法函数?如何调用父类的构造方法?

- 1.构造方法的重载,针对不同的参数个数或类型创建类的对象;
- 2.`this();` 3.`super();`

5-7 阅读下面的程序并写出程序的执行结果,并说明为什么.

```
class S1{
    public static void main(String args[])
    {
        new S2();
    }
    S1()
    {
        System.out.println("S1");
    }
}
class S2 extends S1{
    S2()
    {
        System.out.println("S2");
    }
}
S1 S2
```

5-8 定义一个类 **MyRectangle** 代表矩形,为矩形定义 **getLength** 方法(获得矩形的长度),**getWidth** 方法(获得矩形的宽度),**setLength**(设置长度),**setWidth** 方法(设置宽度),**getArea** 方法(求矩形的面积),**toString** 方法(显示矩形的格式),为矩形派生出一个子类 **MySquare** 代表正方形,并对 **getArea** 和 **toString** 进行重写.并测试;

```
class Test8{
    public static void main(String args[])
    {
        MySquare a =new MySquare(3.5);
        System.out.println("Area is :"+a.getArea());
    }
}
```

```

        System.out.println(a);
        MyRectangle b = new MyRectangle(12,3);
        System.out.println("Area is :"+b.getArea());
        System.out.println(b);
        b.setWidth(5);
        System.out.println(b);
    }
}
class MySquare extends MyRectangle{
    public MySquare(double s)
    {
        super(s,s);
    }
    public double getArea()
    {
        return super.getArea();
    }
    public String toString()
    {
        return "边长是:"+super.getWidth();
    }
}
public class MyRectangle{
    private double width;
    private double length;
    public MyRectangle(double a,double b)
    {
        width=a;
        length=b;
    }
    public double getLength()
    {
        return length;
    }
    public double getWidth()
    {
        return width;
    }
    public void setLength(double a)
    {
        length=a;
    }
}

```

```

public void setWidth(double b)
{
    width=b;
}
public double getArea()
{
    return width*length;
}
public String toString()
{
    return "长为:"+length+",宽为:"+width;
}
}

```

```

PS D:\learning\StudyNote\java\task\4> cd "d:\learning\StudyNote\java\task\5\" ; if ($?) { javac Test8.java } ; if ($?) { java Test8 }
Area is :12.25
边长是:3.5
Area is :36.0
长为:3.0,宽为:12.0
长为:3.0,宽为:5.0

```

5-9 编写一个类,以实现地址的概念,包阔的属性有"国家","省份","市县","街道","门牌","单位","邮编",定义构造方法设置这些属性,并定义一个方法返回一个寄给该地址的信封的字符串.请从地址类派生出国内,国际两种模式,要求在子类中通过方法重载,返回国际或国内的地址字符串.

```

class Test9
{
    public static void main(String args[])
    {
        NativeAddress a =new
NativeAddress("China","Anhui","shucheng","Walk","520","PostOffice","052
0");

        AbroadAddress b =new
AbroadAddress("America","Dex","Geroge","Upon","345","bank","5432");
        System.out.println(a);
        System.out.println(b);
    }
}

```

```

class Address{
    String country;
    String province;
    String city;
    String street;
    String doorpalte;
    String unit;
    String postnumber;
    public Address(String a,String b,String c,String d,String e,String
f,String h)
    {
        country=a;
        province=b;
        city=c;
        street=d;
        doorpalte=e;
        unit=f;
        postnumber=h;
    }
    public String toString()
    {
        return
postnumber+'\n'+country+province+city+street+doorpalte+unit+'\n';
    }
}
class AbroadAddress extends Address{
    public AbroadAddress(String a,String b,String c,String d,String
e,String f,String h)
    {
        super(a,b,c,d,e,f,h);
    }
    public String toString()
    {
        return
super.country+'\n'+super.province+'\n'+super.city+'\n'+super.street+'\n
'+super.doorpalte+'\n'+super.unit+'\n'+super.postnumber;
    }
}
class NativeAddress extends Address{
    public NativeAddress(String a,String b,String c,String d,String
e,String f,String h)
    {

```

```
F:\D:\learning\studynote\java\task3> cd D:\learning\studynote\java\task3 && java -Xmx50M  
0520  
ChinaAnhuishuchengWalk520PostOffice  
  
America  
Dex  
Geroge  
Upon  
345  
bank  
5432
```