

样题 1. 写出一个类 People, 并由该类做基类派生出子类 Employee 和 Teacher。其中 People 类具有 name、age 两个保护成员变量, 分别为 String 类型、整型, 且具有公有的 getAge 成员函数, 用于返回 age 变量的值。Employee 类具有保护成员变量 empno, Teacher 类有 teano 和 zc 成员变量。

答:

```
class People{
    protected String name;
    protected int age;
    public int getAge()
    {
        return age;
    }
}
class Employee extends People{
    protected String empno;
}
class Teacher extends People{
    protected String teano;
    protected String zc;
}
```

样题 2. 编写一个输出“Hello World”, 用两种方式实现(Application、Applet)。

答:

Application:

```
public class Hello {
    public static void main(String[] args)
    {
        System.out.println("Hello
World");
    }
}
```

Applet:

```
import java.awt.*;
import java.applet.*;
public class HelloA extends Applet{
    public void paint(Graphics g)
    {
        g.drawString("Hello World", 50, 25);
    }
}
```

样题 3. 编写一个输出 applet 实现界面, 并实现在第一文本框中输入一个数后, 按“求绝对值”按钮在第二个文本框中显示该数的绝对值, 按“退出”按钮中断程序运行。

答:

```
import java.awt.*; import
java.awt.event.*; import java.lang.Math;
public class Abs implements
ActionListener{
    Frame f;
    TextField tf1, tf2;
    Button b1, b2;
    public void display() {
        f=new Frame("求绝对值例子");
        f.setSize(220, 150);
        f.setLocation(320, 240);

        f.setBackground(Color.lightGray);
        f.setLayout(new
FlowLayout(FlowLayout.LEFT));
        tf1=new TextField(10);
        tf1.setEditable(true);
        tf2=new TextField(10);
        tf2.setEditable(false);
        f.add(tf1);
        f.add(tf2);
        b1=new Button("求绝对值");
        b2=new Button("退出");
        f.add(b1);
        f.add(b2);
        b1.addActionListener(this);
        b2.addActionListener(this);
        f.addWindowListener(new
WinClose());
        f.setVisible(true);
    }
    public void
actionPerformed(ActionEvent e){
        if(e.getSource()==b1)
        {
            int value=(new
Integer(tf1.getText())).intValue();

            tf2.setText(Integer.toString(Math.a
bs(value)));
        }
        else
        {
            if(e.getSource()==b2)
            {
```

```

        System.exit(0);
    }
}

public static void main(String
arg[]){
    (new Abs()).display();
}

class WinClose extends WindowAdapter{
    public void
windowClosing(WindowEvent e)
    {
        System.exit(0);
    }
}

```

样题 4. 定义一个分数类 Fractor, 使能完成分数的加、减运算。请定义测试主类。要求: Fractor, 的数据成员有: 分子 m:分母 n:分数类的方法成员有:Fractor(...) {...} //定义带两个参数的构造方法, 用于给分子分母初始化

void add(Fractor) {...} //定义加运算
 void sub(Fractor) {...} //定义减运算
 void display() {...} //显示分数, 格式为“m/n”
 int factor(int k, int v) //求两数的最大公约数。

答:

```

import java.lang.*;
public class Fractor{
    int m;int n;
    Fractor(int m,int n)
    {
        this.m=m;
        this.n=n;
    }
    void add(Fractor a)
    {
        m+=a.m;
        n+=a.n;
    }
    void sub(Fractor b)
    {
        m-=b.m;
        n-=b.n;
    }
    void display()
    {

```

```

        System.out.println("m/n="+m+"/"+n);
    }
}

int factor(int k,int v)
{
    if (k>v)
    {
        k=k+v;
        v=k-v;
        k=k-v;
    }
    for(int c=k%v;c>0;c=k%v)
    {
        k = v;
        v = c;
    }
    return v;
}

public static void main(String
arg[]){

```

```

    Fractor f=new Fractor(12,18);
    int x=f.factor(12,18);
    f.display();
    f.add(new Fractor(2,2));
    f.display();
    f.sub(new Fractor(1,1));
    f.display();
    System.out.println("x="+x);
}

```

样题 5. 编写一个 Applet 程序, 从键盘输入一个字符串, 要求按逆序打印各位字符串。

答:

```

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class ReverseString extends Applet
implements ActionListener{
    Frame f;
    TextField tf1,tf2;
    Button b1;
    public void init()
    {
        tf1=new TextField(10);
        tf1.setEditable(true);
        tf2=new TextField(10);

```

```

        tf2.setEditable(false);
        b1=new Button("逆序字符串");
        b1.addActionListener(this);
        add(tf1);
        add(tf2);
        add(b1);
    }
    public void actionPerformed(ActionEvent e){
        if(e.getSource()==b1)
        {
            String str=tf1.getText();
            tf2.setText(RevStr(str));
        }
        else
        {
        }
    }
    public void paint(Graphics g)
    {
    }
    public String RevStr(String mystr){
        int length=mystr.length();
        // System.out.print(length);
        String sTemp="";
        for(int i=length;i>0;i--)
        sTemp+=mystr.substring(i-1,i);
        return sTemp;
    }
}

```

样题 6. 给定一个数组 :int[] arrayOfInts={32, 87, 3, 589, 12, 1076, 2000, 8, 622, 127};请按从小到大的顺序排列, 并打印出来。

答:

```

public class ArrayOfInts{
    public static void main(String
args[]){
        int[]
ArrayOfInts={32,87,3,589,12,1076,2000,
8,622,127};
        int i,j,k;
        for(i=0;i<10;i++)
        {
            System.out.print(ArrayOfInts[i]+"

```

```

");
        }
        for(i=0;i<10;i++)
            for(j=0;j<10-i-1;j++)
                if
                (ArrayOfInts[j]>ArrayOfInts[j+1])
                {
                    k=ArrayOfInts[j];
                    ArrayOfInts[j]=ArrayOfInts[j+1];
                    ArrayOfInts[j+1]=k;
                }
                System.out.println("\n排序后的结果为: ");
                for(i=0;i<10;i++)
                {
                    System.out.print(ArrayOfInts[i]+"
");
                }
            }
        }
    }
}

```

样题 7. 编写一个 Java Application 程序, 求 1!+2!+3!+...+20!

答:

```

public class JieCheng{
    public static long GetJieCheng(long
n){
        if (n<=0)
            return 1;
        else
            return n*GetJieCheng(n-1);
    }
    public static void main(String
args[]){
        long s=0;
        for(int i=1;i<=20;i++)
        {
            s+=GetJieCheng(i);
        }
        System.out.println(s);
    }
}

```

样题 8. 编写程序, 完成下面功能:

1) 设计一个 Array 类, 添加一个整型数组作为其数据成员, 添加构造方法以对数组赋初值。

2) 为 Array 类添加数组的求和方法, 添加返回求和值的方法。

3) 编写 Application 程序利用 Array 计算数组的求和值并输出。

答:

```
import java.util.Arrays.*;
public class ArrayP{
    int[] Array;
    ArrayP(int n){
        Array=new int[5];
        for(int i=0;i<5;i++)
            Array[i]=i+1;
    }
    int Anum(int n)    {
        int s=0;
        for(int i=0;i<n;i++)
            s+=Array[i];
        return s;
    }
    public static void main(String
args[]){
        ArrayP p=new ArrayP(5);
        System.out.println(p.Anum(5));
    }
}
```

样题9. 应用 AWT 包编写一个 GUI 程序, 运行后有一个界面, 该界面包括两个文本框, 两个按钮, 一个标签。要求程序实现如下功能: 当用户在两个文本框中输入两上整数后, 鼠标点击平均值按钮以后, 在标签里显示这两个数的平均值; 鼠标点击“最大值”按钮后, 在标签里显示这两个数的较大的那个数。

答:

```
import java.awt.*;
import java.awt.event.*;
import java.lang.Math;
public class MaxAverage implements
ActionListener{
    Frame f;
    TextField tf1,tf2;
    Button b1,b2;
    Label l1;
    public void display(){
        f=new Frame("求绝对值例子");
        f.setSize(260,150);
        f.setLocation(320,240);
```

```
        f.setBackground(Color.lightGray);
        f.setLayout(new
FlowLayout(FlowLayout.LEFT));
        tf1=new TextField(10);
        tf1.setEditable(true);
        tf2=new TextField(10);
        tf2.setEditable(true);
        f.add(tf1);
        f.add(tf2);
        l1=new Label("结果显示在这里: ");
        f.add(l1);
        b1=new Button("平均值");
        b2=new Button("最大值");
        f.add(b1);
        f.add(b2);
        b1.addActionListener(this);
        b2.addActionListener(this);
        f.addWindowListener(new
WinClose());
        f.setVisible(true);
    }
    public void
actionPerformed(ActionEvent e){
        if(e.getSource()==b1)
        {
            try
            {
                float value1=(new
Integer(tf1.getText())).floatValue();
                float value2=(new
Integer(tf2.getText())).floatValue();
                float
value=(value1+value2)/2;
                l1.setText(Float.toString(value));
            }
            catch(Exception e1)
            {
                System.out.println("exception:"+e1.
getMessage());
                e1.printStackTrace();
                System.exit(0);
            }
        }
        else
```

```

        {
            if(e.getSource()==b2)
            {
                try
                {
                    int value1=(new
Integer(tf1.getText()).intValue();
                    int value2=(new
Integer(tf2.getText()).intValue();

                    value1=value1>value2?value1:value2;

                    l1.setText(Integer.toString(value1)
);
                }
                catch(Exception e1)
                {

                    System.out.println("exception:"+e1.
getMessage());

                    e1.printStackTrace();
                    System.exit(0);

                }

            }

        }

        public static void main(String
argv[]){
            (new MaxAverage()).display();
        }
    }

    class WinClose extends WindowAdapter{
        public void windowClosing(WindowEvent e)
        {
            System.exit(0);
        }
    }
}

```

样题 10. 定义一个 Point 点类，该类具有 x,y(表示点的横、纵坐标)两个 float 类型的属性，并定义两个构造方法，一个无参数，将 x,y 均设置为 0，另一个以坐标值为参数，设置 x,y 为给定坐标值。该类的 show 方法输出该点的坐标值。

答：

```

public class Point{
    private float x;

```

```

    private float y;
    Point ()
    {
        x=0;
        y=0;
    }
    Point (int x,int y)
    {
        this.x=x;
        this.y=y;
    }
    void show()
    {
        System.out.println("x="+x+"
y="+y);
    }

    public static void main(String
argv[]){
        Point p=new Point();
        p.show();
        Point p1=new Point(3,5);
        p1.show();
    }
}

```

11. 编写程序，完成下面的功能。

1. 编写一个抽象类 Shape，其中有抽象方法 GetArea() 和 GetPerimeter()。

2. 在 Shape 类的基础上派生出长方形类 Rectangle(属性：长、宽)和圆类 Circle(属性：半径)，然后分别实现其计算面积的方法 GetArea() 以及计算周长的方法 GetPerimeter()。

答：

```

abstract class Shape {
    public abstract double GetArea();
    public abstract double GetPerimeter();
}

class Rectangle extends Shape{
    private double length;
    private double width;
    Rectangle(){length=0;width=0;};
    Rectangle(double length,double
width)
    {
        this.length=length;
        this.width=width;
    }
}

```

```

    public double GetArea()
    {
        return length*width;
    }
    public double GetPerimeter()
    {
        return (length+width)*2;
    }
}
class Circle extends Shape{
    private double r;
    Circle(){r=0;};
    Circle(double r)
    {
        this.r=r;
    }
    public double GetArea()
    {
        return Math.PI*r*r;
    }

    public double GetPerimeter() {
        return 2*Math.PI*r;
    }
}
class AbstractShape{
    public static void main(String
args[]){
        Rectangle rec=new Rectangle(3,4);
        double rec_area=rec.GetArea();
        double
rec_peri=rec.GetPerimeter();
        System.out.println("Rectangle's
area is:"+rec_area+" "+"Rectangle's
perimeter is:"+rec_peri);
        Circle cir=new Circle(5);
        double cir_area=cir.GetArea();
        double
cir_peri=cir.GetPerimeter();
        System.out.println("Circle's
area is:"+cir_area+" "+"Circle's
perimeter is:"+cir_peri);
    }
}

```