



期末练习题



一、程序阅读（共20分，每题5分）

1、下面程序运行的输出结果是_____。

```
public class Test1_1 {  
    public static void main(String[] args) {  
        int a[] = new int[8];  
        int product = 1;  
        for(int i = 2; i <= a.length + 1; i++)  
            a[i-2] = i;  
        for(int i = 0; i < a.length; i++) {  
            if(a[i] % 2 == 1)  
                product *= a[i];  
        }  
        System.out.println(product);  
    }  
}
```



一、程序阅读（共20分，每题5分）

2、下面程序运行的输出结果是_____。

```
public class Test1_2 {  
    public static void printNum(int n) {  
        int digit = n % 10;  
        System.out.print(digit);  
        if(n >= 10)  
            printNum(n / 10);  
    }  
    public static void main(String[] args) {  
        int x = 405;  
        printNum(x);  
    }  
}
```

3、下面程序运行的输出结果是

。

```
class Vehicle {  
    String brand = "通用品牌";  
    static String category = "交通工具";  
    public void start() {  
        System.out.println(brand + " 启动了");  
    }  
    public void info() {  
        System.out.println(category + ":" + brand);  
    }  
}  
  
class Car extends Vehicle {  
    String brand = "宝马";  
    static String category = "汽车";  
    public void start() {  
        System.out.println(brand + "车启动");  
    }  
}
```

```
    public void showDetails() {  
        System.out.println(this.brand + ":" + super.brand);  
        System.out.println(category + ":" +  
Vehicle.category);  
    }  
}  
  
public class Test1_3 {  
    public static void main(String[] args) {  
        Vehicle v = new Car();  
        v.start();  
        v.info();  
        Car c = (Car) v;  
        c.showDetails();  
    }  
}
```

一、程序阅读（共20分，每题5分）

4、下面程序运行的输出结果是_____。

阅读材料：ArrayDeque<E>类。

1) ArrayDeque<E>类创建的对象是一个双端队列（deque, double-ended queue），可以在队列的两端进行插入和删除操作。ArrayDeque基于可变数组实现，比LinkedList作为队列使用时性能更好。

2) ArrayDeque<E>类提供了以下双端队列操作方法：

public void addFirst(E e)：在队列头部插入元素

public void addLast(E e)：在队列尾部插入元素

public E removeFirst()：移除并返回队列头部元素，如果队列为空会抛出异常

public E removeLast()：移除并返回队列尾部元素，如果队列为空会抛出异常

public int size()：返回队列中元素的个数

public boolean isEmpty()：判断队列是否为空

3) ArrayDeque对象的遍历方法：

可以使用增强型for循环从头到尾遍历队列元素。举例：

```
for(E element : arrayDeque) System.out.println(element);
```



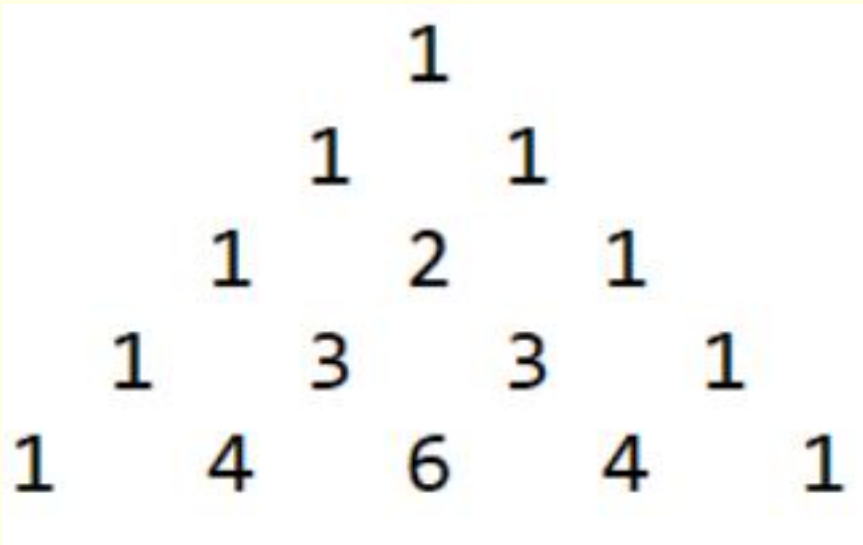
一、程序阅读（共20分，每题5分）

题号	答案	题号	答案
(1)	945（答对得5分）	(2)	504（答对得5分）
(3)	宝马车启动（2分） 交通工具:通用品牌（1分） 宝马:通用品牌（1分） 汽车:交通工具（1分）	(4)	队列内容: ECABD（2分） DE（2分） 剩余元素个数: 3（1分）

```
import java.util.ArrayDeque;
public class Test1_4 {
    public static void main(String[] args) {
        String data = "A-B-C-D-E";
        String[] items = data.split("-");
        ArrayDeque<String> deque = new ArrayDeque<String>();
        for(int i = 0; i < items.length; i++) {
            if(i % 2 == 0) {
                deque.addFirst(items[i]);
            } else {
                deque.addLast(items[i]);
            }
        }
        System.out.print("队列内容: ");
        for(String item : deque) {
            System.out.print(item);
        }
        System.out.println();
        System.out.println(deque.removeLast() + deque.removeFirst());
        System.out.println("剩余元素个数: " + deque.size());
    }
}
```

二、程序填空（共30分，每空2分）

1、以下程序可以输出杨辉三角的前五行，其运行界面如下图所示。请阅读程序并完成填空。





```
public class Test2_1 {  
    public static void main(String[] args) {  
        int n=6;  
        int (1) =new int[n][n];  
        a[1][1]=1;  
        for(int i=2;i<n;i++){  
            a[i][1]=1;  
            (2) =1;  
            for(int j=1;j<i;j++){  
                a[i][j]=a[i-1][j-1]+ (3) ;  
            }  
        }  
    }  
}
```

```
      1  
    1  1  
  1  2  1  
1  3  3  1  
1  4  6  4  1
```

```
int dis=8;  
for(int i=1;i<n;i++){           //显示杨辉三角  
    for(int k=1; (4) ; k++){  
        System.out.print(" ");           //引号中有1个空格  
    }  
    for(int j=1;j<=i;j++){  
        System.out.print(a[i][j]+" "); //引号中有3个空格  
    }  
    System.out.println();  
    (5) ;  
}  
}  
}
```


2.以下Java程序使得display方法能够正确输出局部变量、内部类成员变量、子类成员变量以及父类成员变量的温度数值。程序最终输出为：

19.8

32.0

38.5

25.0

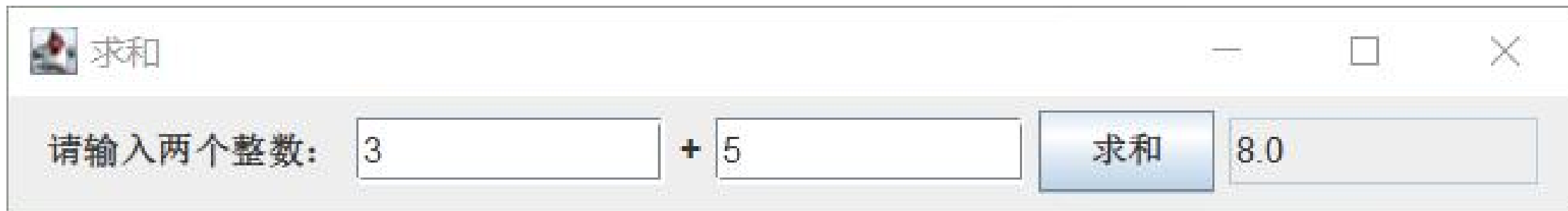
请阅读以下程序并完成填空。



```
class Climate {  
    float temperature = 25.0f;  
    // 父类Climate的温度  
}  
  
class City (6) Climate {  
    // 继承父类  
  
    private float temperature = 38.5f;  
    // City类的温度  
  
    public class Sensor {  
        private float temperature = 32.0f;  
        // Sensor内部类的温度
```

```
        public void display() {  
            float temperature = 19.8f;           // 局部变量  
            System.out.println( (7) );           // 局部变量  
            System.out.println( (8) );           // 内部类成员  
            System.out.println( (9) );           // 外部类成员  
        }  
    }  
  
    public void display() {  
        System.out.println( (10) );           // 父类成员  
    }  
}  
  
public class Test2_2 {  
    public static void main(String[] args) {  
        City.Sensor sensor = new City().new Sensor();  
        sensor.display();  
        City city=new City();  
        city.display();  
    }  
}
```

3.以下程序运行的界面如下图所示。请阅读程序并完成填空。



空格号	答案	空格号	答案
(1)	a[][]	(2)	a[i][i]
(3)	a[i-1][j]	(4)	k<=dis
(5)	dis=dis-2	(6)	extends
(7)	temperature	(8)	this.temperature
(9)	City.this.temperature	(10)	super.temperature
(11)	swing	(12)	AddFrame()
(13)	false	(14)	FlowLayout()
(15)	setText		

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class AddFrame extends JFrame implements ActionListener {
    private JLabel jl1,jl2; //两个JLabel对象，提示用户输入和输出
    private JTextField jtf1,jtf2,jtf3; //三个JTextField对象，用于用户输入和输出
    private JButton jbAdd;//一个JButton对象，用于用户的点击操作

    public AddFrame() {
        //控件的创建

        jl1 = new JLabel("请输入两个整数：");
        jtf1 = new JTextField(10);
        jl2 = new JLabel("+");
        jtf2 = new JTextField(10);
        jbAdd = new JButton("求和");
        jbAdd.addActionListener(this);//给按钮添加事件响应
        jtf3 = new JTextField(10);
        jtf3.setEditable(false);//结果文本框不能被编辑
    }
}
```

```
add(jl1);
add(jtf1);
add(jl2);
add(jtf2);
add(jbAdd);
add(jtf3);
setLayout(new ____ ( 14 ) );//窗口流式布局
setSize(458, 95);
setTitle("求和");
setLocationRelativeTo(null);
setDefaultCloseOperation(EXIT_ON_CLOSE);
setVisible(true);
}
```

```
//当按钮被点击时执行actionPerformed方法
public void actionPerformed(ActionEvent e) {
    try{
        double d1 = Double.parseDouble(jtf1.getText());
        double d2 = Double.parseDouble(jtf2.getText());
        jtf3.__(15)__("+(d1+d2));
    }catch(Exception ex){
        JOptionPane.showMessageDialog(this, "提示:只能输入两个数字,不能为空或者其他字符");
    }
}

public static void main(String[] args) {
    new AddFrame();//启动窗口
}
}
```



三、编程题 (10分)

编写一个程序，按用户输入的行数打印沙漏图形。运行示例如下图所示。

请输入要打印的行数：

5

*

```
import java.util.Scanner;
```

```
public class Test1 {  
    public static void main(String[] args) {  
        System.out.println("请输入要打印的行数： ");    2分  
        Scanner s = new Scanner(System.in);  
        int rows = s.nextInt();                            2分  
        // 上半部分  
        for (int i = rows; i >= 1; i--) {                  4分  
            for (int j = 1; j <= rows - i; j++) {  
                System.out.print(" ");  
            }  
            for (int k = 1; k <= 2 * i - 1; k++) {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
        // 下半部分    2分  
        for (int i = 2; i <= rows; i++) {  
            for (int j = 1; j <= rows - i; j++) {  
                System.out.print(" ");  
            }  
            for (int k = 1; k <= 2 * i - 1; k++) {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
    }  
}
```

四、编程题 (20分)

- 编写程序，要求如下：
- (1) 编写方法：static void initSortArray(int[] arr)，初始化数组arr中的每个元素为1~100之间的随机整数，并对数组元素按从小到大的顺序进行排序。
- (2) 编写方法：static void printArray(int[] arr, boolean isASC)，根据isASC的取值对已经排序（从小到大）的数组进行打印。如果isASC为true，则升序打印，否则逆序打印。
- (3) 编写方法：static int[] insertArray(int[] arr,int num)，将num值插入到已经排序（从小到大）的数组arr中，保证数组依然有序，并返回新的数组。
- (4) 在main方法中，定义数组并调用initSortArray方法进行初始化和排序，显示排序后的数组元素。生成一个随机整数，并调用insertArray方法将其插入到数组中，打印插入数据后的数组元素。程序运行示例如下：

初始数组：

3	10	20	48	66	67	82	88
---	----	----	----	----	----	----	----

插入数据 21 后的数组：

3	10	20	21	48	66	67	82	88
---	----	----	----	----	----	----	----	----



四、编程题 (20分)

- 编写程序，要求如下：
- (1) 编写方法：static void initSortArray(整数，并对数组元素按从小到大的顺序进行
- (2) 编写方法：static void printArray(int小到大)的数组进行打印。如果isASC为tr
- (3) 编写方法：static int[] insertArray(ir数组arr中，保证数组依然有序，并返回新的
- (4) 在main方法中，定义数组并调用initS生成一个随机整数，并调用insertArray方法行示例如下：

初始数组：

3 10 20

插入数据 21 后的数组：

3 10 20

四、编程题 (20分)
import java.util.Arrays;
public class Test2 {

```
static void initSortArray(int[] arr) {           5分
    for(int i =0;i<arr.length;i++) {
        arr[i]= (int)(Math.random()*100)+1;
    }
    Arrays.sort(arr);
}
static void printArray(int[] arr, boolean isASC) {           4分
    if(isASC) { //由小到大
        for(int i=0;i<arr.length;i++)
            System.out.print(arr[i]+"\\t");
    }else
        for(int i=arr.length-1;i>=0;i--)
            System.out.print(arr[i]+"\\t");
    System.out.println();
}
static int[] insertArray(int[] arr,int num) {           6分
    int[] newarr = new int[arr.length+1];
    boolean flag = false;
    for(int i=0,j=0;i<=arr.length;i++) {
        if(arr[j]<num||flag)
            newarr[i] = arr[j++];
        else {
            newarr[i]=num; flag = true;
        }
    }
    return newarr;
}
public static void main(String[] args) {           5分
    int[] arr = new int[8];
    initSortArray(arr);
    System.out.println("初始数组: ");
    printArray(arr,true);
    int num =(int)(Math.random()*100);
    int[] newarr = insertArray(arr,num);
    System.out.println("插入数据 "+num+" 后的数组: ");
    printArray(newarr,true);
}
```

五、编程题（20分）

- 请根据面向对象的编程思想，按如下要求实现代码：
- 1. 定义一个接口 Device，包含以下抽象方法：
 - (1) void turnOn(): 打开设备。
 - (2) void turnOff(): 关闭设备。
- 2. 定义一个抽象类 AbstractDevice，实现 Device接口：
 - (1) 定义受保护的成员：deviceName(设备名称，String类型)，isOn(设备状态，boolean类型)。
 - (2) 定义带一个参数的构造方法，初始化deviceName属性，isOn的默认值为false。
 - (3) 实现turnOn()方法：将isOn属性设为true，并显示该设备已打开。
 - (4) 实现turnOff()方法：将isOn属性设为false，并显示该设备已关闭。
 - (5) 定义抽象方法：public abstract void performAction()。

五、编程题（20分）

- 3. 定义SmartLight类，继承自AbstractDevice类：
 - (1) 定义私有的成员：brightnessLevel(灯的亮度，int类型)。
 - (2) 定义带有一个参数的构造方法，初始化deviceName属性，brightnessLevel的默认值为0。
 - (3) 定义brightnessLevel 的set方法：public void setBrightness(int level)，如果level在0~100之间，设置灯的亮度为level值，否则显示“亮度级别无效”。
 - (4) 实现performAction()方法：显示设备正在调整亮度，并设置亮度为50。
 - (5) 重写turnOn()方法：在将isOn设为true的同时，设置灯的亮度为100。
 - (6) 重写turnOff方法：在将isOn设为false的同时，设置灯的亮度为0。
- 在测试类的主方法中定义如下代码，程序运行结果如右图所示：

五、编程题 (20分)

```
public class Test3 {  
    public static void main(String[] args) {  
        SmartLight livingRoomLight = new SmartLight("客厅灯");  
        livingRoomLight.turnOn();  
        livingRoomLight.performAction();  
        livingRoomLight.setBrightness(75);  
        livingRoomLight.turnOff();  
    }  
}
```

客厅灯	已打开
客厅灯	亮度设置为: 100%
客厅灯	正在调整亮度...
客厅灯	亮度设置为: 50%
客厅灯	亮度设置为: 75%
客厅灯	已关闭
客厅灯	亮度设置为: 0%

五、编程题（20分）

- 请根据面向对象的编程思想，按如下要求实现代码：
- 1. 定义一个接口 Device，包含以下抽象方法：
- (1) void turnOn(): 打开设备。
- (2) void turnOff(): 关闭设备。

//设备接口

```
interface Device {  
    void turnOn(); // 打开设备  
    void turnOff(); // 关闭设备  
}
```

2分

五、编程题（20分）

- 2. 定义一个抽象类 AbstractDevice，实现 Device接口：
- (1) 定义受保护的成员：deviceName(设备名称，String类型)，isOn(设备状态，boolean类型)。
- (2) 定义带一个参数的构造方法，初始化deviceName属性，isOn的默认值为false。

//抽象设备类，实现了SmartDevice接口

```
abstract class AbstractDevice implements Device {  
    protected String deviceName;  
    protected boolean isOn;  
    public AbstractDevice(String deviceName) {  
        this.deviceName = deviceName;  
        this.isOn = false;  
    }  
}
```

4分

五、编程题 (20分)

- 2. 定义一个抽象类 AbstractDevice, 实现 Device接口:
- (3) 实现turnOn()方法: 将isOn属性设为true, 并显示该设备已打开。
- (4) 实现turnOff()方法: 将isOn属性设为false, 并显示该设备已关闭。
- (5) 定义抽象方法: public abstract void performAction()。

```
public void turnOn() {  
    isOn = true;  
    System.out.println(deviceName + " 已打开");  
}
```

3分

```
public void turnOff() {  
    isOn = false;  
    System.out.println(deviceName + " 已关闭");  
}
```

// 抽象方法, 子类必须实现

```
public abstract void performAction();  
}
```

1分

五、编程题 (20分)

- 3. 定义SmartLight类，继承自AbstractDevice类：
 - (1) 定义私有的成员：brightnessLevel(灯的亮度，int类型)。
 - (2) 定义带有一个参数的构造方法，初始化deviceName属性，brightnessLevel的默认值为0。
 - (3) 定义brightnessLevel 的set方法：public void setBrightness(int level)，如果level在0~100之间，设置灯的亮度为level值，否则显示“亮度级别无效”。

```
class SmartLight extends AbstractDevice { //具体设备类：智能灯    3分
    private int brightnessLevel;
    public SmartLight(String deviceName) {
        super(deviceName);
        this.brightnessLevel = 0;
    }
    public void setBrightness(int level) {                                3分
        if (level >= 0 && level <= 100) {
            brightnessLevel = level;
            System.out.println(deviceName + " 亮度设置为: " + level + "%");
        } else {System.out.println("亮度级别无效，请输入0到100之间的值");}
    }
}
```