

江西师范大学计算机科学技术专业

17-18 第 2 学期《面向对象程序设计》(262194)

期末考试·试题 A

一、简答题 (每小题 10 分 , 共计 30 分)

1、简述结构化程序设计方法和面向对象程序设计方法的核心思想。

答:【结构化程序设计方法】将软件系统视为一组功能的集成。“结构化”是指大功能模块由小功能模块组合而成。各模块独立性较强,以便灵活组合出新模块。设计策略为对功能进行自顶向下逐步求精。

【面向对象程序设计方法】将软件系统视为现实世界(需求)的仿真(或模拟)。现实世界由一组对象组成,系统的功能表现为一组对象间的交互。总体设计策略为:从现实世界(需求)中抽象出一组类和对象,编程实现这组类和对象,进而实现对象间的交互(即对象间的消息传递)。

2、简单说明 java 的异常处理策略。

答: java 将程序运行期间的所有异常均打包成异常对象,异常处理就是对该异常对象的捕获和处理。处理机制包括声明原则和处理原则,前者是指声明自己将抛出何种异常,即当异常发生时,自己不处理,将异常对象交给调用者;而处理原则是指用 try-catch 语句对异常对象的捕获并处理。

3、java 最初设计时,为何十分重视可靠性,并为此实施了哪些举措?

答: Java 极为重视程序的可靠性。这是因为普通程序交付使用后如果发现错误,修改后再交给用户即可。嵌入式程序需要烧制到芯片中。若烧制后发现错误,只有收回产品,更换芯片,代价极高。

Java 从语言层面制定了许多举措以支持可靠性,如强类型限制、语言简单规范、内存自动管理、沙箱机制等。这些举措,让 java 程序在可靠性方面具有先天优势。

二、包及权限的配置 (本题 10 分)

1、a): b1、b2、b3

b): a1、a2、a3

- c): 不能访问 B 类对象的属性, 只能访问 A 中的 a1, 其余均不能访问。
- d): 不能访问 B 类对象的属性, 只能访问 A 中的 a1、a3, 其余均不能访问。

三、基础设计题 (每小题 10 分, 共计 20 分)

1、给定单链表类 LinkedList, 请根据要求补充完成相关序列化和反序列化代码。其中 App 类中的有 (1)、(2)、(3) 处有三段代码需要补充。

(1) implements Serializable

/* 请将链表 L 借助序列化机制写入文件 L.dat */

(2) FileOutputStream fo=new FileOutputStream("L.dat");
 ObjectOutputStream obj_o=new ObjectOutputStream(fo);
 obj_o.writeObject(L); obj_o.close();

/* 请借助反序列化机制将文件 L.dat 中的链表复原, 表头名称为 newL */

(3) FileInputStream fi=new FileInputStream("L.dat");
 ObjectInputStream obj_i=new ObjectInputStream(fi);
 LinkedList new_L=(LinkedList)obj_i.readObject();
 obj_i.close();

2、给定如下类 (不完整)。请设计 "空军" 接口, 并将类补充完整, 以实现 "防空炮可以攻击所有飞行类兵种" 这一需求

interface 空军{;}

class 飞行兵 implements 空军{;}

class 飞艇 implements 空军{;}

class 飞禽 implements 空军{;}

class 防空炮{

public void attack(空军 k){
 String s=k.getClass().getName();
 System.out.print("攻击 "+s+"\n");
 }

}

class Exam_3{

public static void main (String[] args) {
 防空炮 fkp=new 防空炮();
 空军[] k={new 飞禽(), new 飞艇(), new 飞行兵()};

```

        for( 空军 x: k )
            fkp.attack(x);
    }
}

```

四、综合设计题 (每小题 20 分 , 共计 40 分)

1. 给定如下不完全的 m 度树类 Tree,

a. (5 分)

```

void pre(){
    System.out.print(this.data+" ");
    for(int i=0; i<m; i++)
        if(this.c[i]!=null)
            this.c[i].pre();//对自己 c[i]号孩子进行遍历
}

```

b. (7 分)

```

class Queue{
    private final int max=10;
    private Tree[] data=new Tree[max];
    private int f,r;
    public void add(Tree t)//入队
        { data[r]=t; r=(r+1)%max;}
    public Tree out() //出队
        { int x=f; f=(f+1)%max; return data[x];}
    public boolean isEmpty(){ return r==f; }//判空
}

```

c. (8 分)

```

void level(){Tree t;
    Queue q=new Queue();
    q.add(this);
    while(!q.isEmpty()){
        t=(Tree)q.out();
        System.out.print(t.data+" ");
        for(int i=0; i<m; i++)
            if(t.c[i]!=null)q.add(t.c[i]);
    }
}

```

2、第一处补充: 5 分

```

public GGQ(String[]d, int m){data=d; max=m;}

```

```
private synchronized int getPos(){
    if(pos==max)pos=0;
    else pos++;
    return pos;
}
```

第二处补充: 10 分

```
public FaBuZhe(String na, String[] d, GGQ g)
    {name=na; data=d;ggq=g; t=new Thread(this);}
public void start(){t.start();}
public void join(){
    try{t.join();}    catch(Exception e){;}
}
```

第二处补充: 5 分

```
FaBuZhe f1=new FaBuZhe("王大",a,g);
FaBuZhe f2=new FaBuZhe("王二",b,g);
FaBuZhe f3=new FaBuZhe("王三",c,g);
f1.start();f2.start(); f3.start();
f1.join();f2.join();f3.join();
```

江西师范大学计算机科学技术专业

17-18 第 2 学期《面向对象程序设计》(262194)

期末考试·试题B

一、简答题 (每小题 10 分 , 共计 30 分)

1、简述结构化程序设计方法和面向对象程序设计方法的核心思想。

答:【结构化程序设计方法】将软件系统视为一组功能的集成。“结构化”是指大功能模块由小功能模块组合而成。各模块独立性较强,以便灵活组合出新模块。设计策略为对功能进行自顶向下逐步求精。

【面向对象程序设计方法】将软件系统视为现实世界(需求)的仿真(或模拟)。现实世界由一组对象组成,系统的功能表现为一组对象间的交互。总体设计策略为:从现实世界(需求)中抽象出一组类和对象,编程实现这组类和对象,进而实现对象间的交互(即对象间的消息传递)。

2、假定期望设计一个类 A , 类中包含一个必须要拥有 public 属性的方法 f()。当不希望其他程序员重写此方法时,请举出三种处理策略?

答: a. 给 f()方法施加 static 修饰; b. 给 f()方法施加 final 修饰; c. 将类 A 设定为 final 类。

3、简述 IO 机制的核心思想、处理策略及其优点

答:流是对象间数据交互的统一形式。流是单向的,输入流配备读操作(即只能读);输出流配备写操作(即只能写)。应用策略类似生产流水线,即按需组合。这样,在使用流时,无需关心数据交互的另一端是什么对象,或是在什么位置,只需根据需从流中读取或向流写入即可。

二、包及权限的配置 (本题 10 分)

1、a): b1、b2、b3

b): a1、a2、a3

c): 不能访问 B 类对象的属性,只能访问 A 中的 a1,其余均不能访问。

d): 不能访问 B 类对象的属性,只能访问 A 中的 a1、a3,其余均不能访问。

三、基础设计题 (每小题 10 分 , 共计 20 分)

1、

(1)、

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

(2)、

```
implements ActionListener
```

(3)、

```
    b_exit.addActionListener(this);  
    b_count.addActionListener(this);
```

(4)、

```
    public void actionPerformed(ActionEvent e){  
        if (e.getSource()==b_exit) System.exit(0);  
        if (e.getActionCommand().equals("计数")) {  
            count++;  
            cLa.setText("您点击了"+ count+"次");  
            setVisible(true);  
        }  
    }  
}
```

2、创建一个 **SuperMan** 类, 该类至多能构造出 4 个对象。要求: 为使类更易于维护, 必须单独设计一个 **boolean limit(...)** 函数, 实现对数量限制的检测, 满足数量限制则返回真, 否则返回假。c. 设计该 **SuperMan** 类, 其中包含构造 **SuperMan** 类对象的手段。

```
class SuperMan{  
    private static int count;  
    private SuperMan(){ count++; }  
    private static boolean limit(){  
        if(count<4) return false;  
        return true;  
    }  
    public static SuperMan creatSuperMan(){  
        if(limit()==true) return new SuperMan();  
        return null;  
    }  
}
```

```

    }
}

```

四、综合设计题 (每小题 2 分 , 共计 40 分)

1. 给定如下不完全的二叉树类 BinTree,

a. (5 分)

```

public void inOrder(){//中序遍历

    if(L!=null)L. inOrder ();
    System.out.print(data+" ");
    if(R!=null) R. inOrder ();

}

```

b. (8 分)

```

class Stack{
    private final int max=20; //栈的最大容量
    private Bintree[] data;
    private int top;
    public boolean isEmpty(){ return top==0; }
    public void push(BinTree x){ data[top]=x; top++; }
    public BinTree pop(){ //出栈
        if(top==0)return null;
        top--; return data[top];
    }
}

```

c. (7 分)

```

void preOrder() {
    Stack st=new Stack();
    BinTree t=this;
    while(t!=null||!st.isEmpty())
        if(t!=null){
            st.push(t);
            System.out.print(t.data+" ");
            t=t.L;
        }
        else{ t=st.pop(); t=t.R; }
}

```

2、下面代码模拟实现 3 台 PC 机共享打印的情形。。下面以给出部分代码，请按照要求补充完整。（20 分）

2、第一处补充：15 分

```
public PC(String f,String n, String []d)
    { printFlag=f;name=n; neiRong=d; t=new Thread(this); }
public void start(){t.start();}
public void join(){
    try{t.join();}
    catch(Exception e){;}
}
public void run(){
    synchronized(printFlag){
        System.out.print(name+": ");
        for(int i=0; i<neiRong.length; i++){
            System.out.print("\t"+neiRong[i]);
            try{Thread.sleep(10); }
            catch(InterruptedException e){;}
        }
        System.out.println();
    }
}
```

第 2 处补充：5 分

```
PC t1=new PC(s,"作业 1",s1);
PC t2=new PC(s,"作业 2",s2);
PC t3=new PC(s,"作业 3",s3);
t1.start (); t2.start (); t3.start ();
t1.join();t2.join();t3.join();
```