

# 江西师范大学计算机科学技术专业

## 15-16 第 2 学期《面向对象程序设计》(262194)

### 期末考试·试题 A

#### 一、基础知识题 (每小题 10 分, 共计 30 分)

1、类 x 将输出二维数组中的所有数据, 请将其补充完成。

```
class X{
    public static void main (String[] args) { int i,j;
        int[][]a={{1,2,3},{4,5},{6,7,8,9}};
        for(i=0; i<a.length; i++){
            for(j=0; j<a[i].length; j++)
                System.out.print(a[i][j]+" ");
            System.out.print("\n");
        }
    }
}
```

输出结果为:

```
1 2 3
4 5
6 7 8 9
```

2、下面程序将输出 A、B 中的私有属性数据。请为类 A、B 补充一些方法, 使其能得到给定的输出结果。注: 不得改动给定部分。

```
class A{
    private int x;
    public A(int x){this.x=x;}
    public int getX(){return x;}
}
class B extends A{
    private int y;
    public B(int x,int y){super (x); this.y=y;}
    public void showInfo(){
        System.out.print("x= "+getX()+" y="+y);
    }
    public static void main(String[] args) {
        new B(3,5).showInfo();
    }
}
```

输出结果为:

```
x= 3 y=5
```

3、下列代码用线程模仿共享打印, 打印内容见 main() 中的 s1、s2、s3。要求: 每个

人的打印内容必须完整，不得交叉。这里的输出结果仅是其中一种可能。

```
public void run(){  
    synchronized(data){  
  
        System.out.print(name+"说: ");  
  
        for(int i=0; i<say.length; i++)  
            System.out.print(say[i]);  
        System.out.println ();  
    }  
}
```

## 二、简答题（每小题 10 分，共计 30 分）

1、简单说明编译器、字节码、虚拟机在 java 实现跨平台过程中起的作用。

答：

编译器：与平台有关，将源码翻译成统一的字节码；

字节码：java 对源码的统一解释；

虚拟机：与平台相关，基于自身平台的指令系统，解释执行字节码。

2、简述面向对象程序设计方法的核心思想。

答：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）。

3、为何说大程序开发的侧重点是可靠性、可维护性和可重用性？

答：a. 由于大程序很难确保完全正确，关系少量的核心模块可靠更为现实；

b. 由于维护活动涉及软件从设计、编码到使用、维护整个过程，且频繁发生。易于维护（即可维护性）的设计，将能有效降低代码的修改难度，减少错误发生的机率，故备受重视；

c. 模块易于重用，不仅可减少开发工作量，而且重用的模块往往更可靠。

## 三、综合设计题（每小题 10 分，共计 40 分）

1、创建一个 Student 类，包含 name（String 型）、age（int 型）两个私有属性。并满足如下要求：a. 在创建对象时，name 不得为空，且 age 的取值范围为： $5 \leq \text{age} \leq 25$ ，否则将无法创建对象；b. 为使类更易于维护，必须单独设计一个 boolean limit(...) 函数，实现对 name 和 age 的限制检查，符合创建条件则返回真，否则返回假。c. 构造设计该 student 类，其中包含构造 Student 类对象的手段。

```
class Student{  
    private String name;  
    private int age;  
    private Student(String n, int a){ name=n; age=a; }  
    private static boolean limit(String n, int a){  
        if(n==null || a<5 || a>25) return false;  
        return true;  
    }  
}
```

```

    }
    public static Student creatStudent(String n, int a){
        if(limit(n,a)==true) return new Student(n,a);
        return null;
    }
}

```

2、某公司有董事长、经理等不同类型的员工，均具有私有属性：String 型的 name。他们关联同一个 Data 型的数据集，但看到的视图不同。请基于**抽象类**，构造一组类或接口，使之能得到给定的输出结果。其中 App 类中的有（1）、（2）两处空需要填写。

**注意：类中定义的所有属性均只能为私有权限。**

```

class Data{;}
abstract class 员工{
    private String name;
    private Data data;
    public 员工(String s, Data d){name=s; data=d;}
    public String getName(){return name;}
    abstract public void dealData();
}
class 董事长 extends 员工{
    public 董事长(String s, Data d){super(s,d);}
    public void dealData(){System.out.println("董事长"+getName()+"的视图...");}
}
class 经理 extends 员工{
    public 经理(String s, Data d){super(s,d);}
    public void dealData(){System.out.println("经理"+getName()+"的视图...");}
}
class App{
    public static void view(员工 s){ s.dealData(); }
    public static void main (String[] args) {
        Data d=new Data();
        员工 [] s={new 董事长("张三",d),new 经理("李四",d),new 经理("王五",d)};
        for(int i=0; i<s.length; i++)
            view(s[i]);
    }
}

```

3、给定包，有 A、B、C 三个类，分别隶属于 X、Y、Z 三个包。请完善下面的代码框架，使得可以允许在 B 中可以访问 A 中的方法 showInfo()，但 C 不可以。（10 分）

**注 A、B、C 均需要完善，但方法内容不可更改。**

```

package X;
public class A{
    protected void showInfo(){System.out.print("\nAAAA.");}
}
package Y;
import X.A; //=== 缺少此句扣 2 分 --- 只扣一次

```

```

public class B extends A{
    public void visit(){ showInfo(); } //(1)正确
}
package Z;
import X.A; //=== 缺少此句扣 2 分 --- 只扣一次
public class C{
    public void visit(){ new A().showInfo();} //(2)产生编译错
}

```

4、假设某系统设计中已经确定事件源 *s* (*Comp\_1* 类型), *s* 上将产生 *EvEvent* 类型事件, *EvEvent* 型事件对应的监听器接口为 *EvListener*, 该接口中包含两个事件处理方法 *action\_1(EvEvent e)*、*action\_2(EvEvent e)*。*Comp\_1* 类中包含有与 *EvListener* 接口绑定方法: *void addEvListener(EvListener eL)* 设计代码框架实现如下要求: 当 *s* 上发生 *Ev* 型事件并触发 *action\_1()* 动作时, 输出: "action\_1 动作被激发..."。

注: 为确保事件能够正常传播、捕获并激发事件处理动作, 假定上述所有的 *class*、接口、接口中的方法均由系统预定义。另外, 需要指明必须要导入的包。(10 分)

```

import java.awt.*;
import java.awt.event.*;
class GUI extends JFrame implements EvListener{
    GUI(){
        Comp_1 s=new Comp_1();
        //假定界面构造部分已完成
        s.addEvListener(this);
    }
    action_1(EvEvent e){System.out.print("action_1 动作被激发...
");}
    action_2(EvEvent e){ ; }
}

```