2015-2016第二学期面向对象程序设计（理论）模拟卷答案
(很多答案不唯一，仅供参考)

一、基础知识题（每小题 10 分，共计 30 分）

1、类 **X** 将输出 **n** 层的数字三角（第 **i** 行打印 **i** 个 **i**）。请将其补充完成。

注：用打印语句罗列输出结果将不给分。

```
class X{
   public static void main (String[] args) { int i,j;
      for(i=1; i<4; i++){
         for(j=1; j<=i; j++)
            System.out.print( i +"  "); //将 i 写成 j 扣 5 分
         System.out.print("\n");
      }
   }
}
```

输出结果为：

```
1
2 2
3 3 3
```

2、下面程序将交换两个对象的数据。请补充完成类 **Data** 和 **App**，使其能得到给定的输出结果。注：用打印语句罗列输出结果将不给分。

```
class Data{ private int d;
   public Data(int x){d=x;}
   public int getD(){return d;}
}
class App{
   public static void showResult(Data d1, Data d2){
      System.out.print(d1.getD()+"  "+d2.getD());
   }
   public static void switchData(Data d1, Data d2){ //缺少 static  扣 2 分
      int x;
      x=d1.d; d1.d=d2.d; d2.d=x;
   }
   public static void main (String[] args) {
      Data d1=new Data(3); Data d2=new Data(5);
      switchData(d1,d2); //交换两个对象
      showResult(d1,d2); //依次打印 d1、d2 对象中的数据
   }
}
```

输出结果为：

```
5 3
```

3、下列代码用线程演示 **3** 个线程的同步：按 **t1-t2-t3** 这一特定次序依次输出。每个线程每轮只能输出一个数组元素。

```
   public void setData(int x, int y){
```

```
        while(data!=x)
            try{ wait(); }   catch(InterruptedException e){;}
        data=y;   notifyAll();
    }
    public void run(){
        synchronized (flagData){
            for(char c: outputData){
                flagData.setData(runFlag, nextFlag);
                System.out.print("  "+c);
            }
        }
    }
```

二、简答题（每小题 10 分，共计 30 分）

1、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。

2、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）。

3、什么是设计的可维护性，简要说明其对软件设计的重要意义。

答：软件的可维护性是指理解、改正、改动、改进软件的难易程度。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；

三、综合设计题（每小题 10 分，共计 40 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）处空需要填写。

```
（1）implements Serializable
        /* 请将链表 L 借助序列化机制写入文件 L.dat */
        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 */
        FileInputStream fi=new FileInputStream("L.dat");
        ObjectInputStream obj_i=new ObjectInputStream(fi);
```

```
      LinkedList new_L=(LinkedList)obj_i.readObject();
   obj_i.close();
```

2、某云计算平拟向客户提供计算服务策略如下：向用户提供 MyApp 类，该类提供计算服务：void compute()。MyApp 类中还包含私有的用户信息 name，以及验证方法 verify()。客户向云计算平台提交包含自己的用户信息和计算方法。下面程序模拟上述过程，其中客户 A 的计算方法输出信息“我来试试”；客户 B 的计算方法输出信息“到此一游”。请基于**抽象类**，构造一组类，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。

```
class UserInfo{;}//用户信息
abstract class MyApp{
    private UserInfo user;
    public static boolean verify(){/*验证 user 是否为合法用户*/
          return true;
          } //请为 verify 方法添加合适的修饰，使得用户无法更改

   public MyApp(UserInfo u){ user=u;}
   public abstract void compute();
}
class Client_A extends MyApp{
   public Client_A(UserInfo u){ super(u); };
   public void compute( ){System.out.println("到此一游！");}
}
class Client_B extends MyApp{
   public Client_B(UserInfo u){ super(u); };
   public void compute( ){System.out.println("我来试试！");}
}
class App{
   public static void runCompute(MyApp m){ m.compute();}
   public static void main (String[] args) {
      UserInfo u1,u2;      Client_A a; Client_B b;
      u1=new UserInfo();  a=new Client_A(u1);runCompute(a);
      u2=new UserInfo();  b=new Client_B(u2);runCompute(b);
   }
}
```

输出结果为：

到此一游！

我来试试！

3、创建一个三角形类，包含属性：private int a,b,c;分别代表三角形的三条边。设计是需要满足如下需求： a. 在创建对象输入三条边，三边取值必须合法（三边均为正值，且任意两边之和大于第三边），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，实现对三条边的限制检查，符合创建条件则返回真，否则返回假。c.该类有一个 public void setEdges(int x, int y, int z)方法，将 a/b/c 的值替换成 x/y/z。但当 x,y,z 的值不满足三角形限制条件时，将不予替换，直接返回；d.设计该三角形类，其中包含构造三角形类对象的手段。

```
class SanJiao{
   private int a,b,c;
   private SanJiao(int x, int y, int z){ a=x; b=y; z=c;}
   private static boolean limit(int x, int y, int z){
      if(x>0&&y>0&&z>0&&(x+y>z)&&(x+z>y)&&(y+z>x)) return true;
      return false;
   }
   public static SanJiao creatSanJiao(int x, int y, int z){
      if(limit(x,y,z)==true) return new SanJiao(x,y,z);
      return null;
   }
   public void setEdges(int x, int y, int z){
      if(limit(x,y,z)==true) { a=x; b=y; z=c;};
   }
}
```

4、程序运行初始界面如下左图，界面如下图所示。每次点击“计数”按钮，均会在相关标签中给出计数提示；点击退出按钮，则会结束程序。请补充完成此程序。另外，需要指明必须要导入的包，并填写（**1**）处的空

提示：**String.valueOf(10)**可以将 **10** 转换成 **String** 型**"10"**

```
import java.awt.*; import java.awt.event.*; import javax.swing.*;
class MyGUI extends JFrame implements ActionListener{
   private JButton b_count,b_exit;
   private Label cLa;   private int count;
   public MyGUI() {
      count=0;    setSize(300,100); setLayout(new FlowLayout());
       cLa=new Label(" 0 ");            add(cLa);
       b_count = new JButton("计数");      add(b_count);
       b_exit = new JButton("退出");    add(b_exit);
       setVisible(true);
       //以上为 GUI 界面设计部分

       b_exit.addActionListener(this);
       b_count.addActionListener(this);
   }
       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"+String.valueOf(count)+"次");
             setVisible(true);
```

```
        }
    }
    public static void main(String[] args) {    new MyGUI();   }
}
```