

## 由一个简单的程序谈起——之二

江苏 无锡 缪小东

下面就是关于这个程序的一些建议了！肯定不完整，希望大家补上！

第一、肯定选择中文做对象名是不太合适的！

一般不太会使用中文做变量名（虽然 Java 支持 Unicode，可以这么做！）。在界面上使用中文是在所难免的！不过不是通过硬编码的方式，可以使用 `Properties` 和 `Local` 两个类组合，控制界面上的说明性文字！

第二、这个程序里有些做法还是不错的！例如：程序中的很多 `Label` 对象放在构造方法的内部，根本就没有放入成员变量中。例如：`box2.add(new JLabel("姓名:",JLabel.CENTER));` 一般必须限制它们到最小的范围中，就是说其它方法没有使用到的，只在某个方法中出现的变量，可以完全放到该方法中。这样做是为了限制对象的生命周期，同时为了程序的清晰！`JTextField`、`JRadioButton` 都必须作为类的属性、而 `ButtonGroup` 是很不应该作为类的属性的！为什么？自己考虑吧！

第三、基本信息表=`(Hashtable)inTwo.readObject()`;

一次将所有的学生记录全部读出，好像不太好！假如你数据库比较大——学生记录比较多！IO 肯定会阻塞，同时因为它的操作放在 GUI 的事件中，理所当然还导致 GUI 冻结——当进行该事件处理时，GUI 上所有的部件都是不可以操作的，就像冻结了一样！对于这种情况可以将该事件操作放于一个线程中！具体可以研究《Java Thread Programming》吧！

第四、事件处理方法中好几个对象都有 `setEditable(false)` 方法，这些方法都是在同一个逻辑下的操作，因此可以将其放入一个统一的方法中，如：

```
public void setEditable(){
    name.setEditable(false);           //姓名.setEditable(false);
    .....
}
```

这种改法，在本程序中可能不算很好，业务逻辑表明：所有的 `JTextField` 被设置为不可编辑！在 Swing 中主要的模式就是组合模式，我们可以利用

`JPanel.getComponents()` 方法得到所有的 `Component` 数组，然后查看其是不是 `JTextField`，是就 `setEditable(false)`。其它如清空所有的项就可以是 `clear` 方法了！初始化界面也尽量放到一个 `iniGUI` 方法中好了！总之尽量将同一逻辑，相同的方法组织到一起！

第五、其它还有下面这些关于文件的操作其实也不应该放到类的属性的！（我们会选择一些其它方法！）为什么让客户看到你写入文件呢！

```
FileInputStream inOne=null;
ObjectInputStream inTwo=null;
FileOutputStream outOne=null;
ObjectOutputStream outTwo=null;
```

第六、以下是本篇的主要内容！这个程序总体上是 GUI 和业务逻辑完全混合，以后调试、扩展可能相当费力！下面就简单地讲述吧！

在我们的程序里，很显然要对文件进行操作。这些操作可以直接发到某一个对象中，只是可以使用一个 DOA（Data Object Access 的模式），将数据存取和 GUI 分开！程序中使用 `File` 作为持久层，不能以后会采用数据库作为持久层，这中变化是会有有的，因此根据《Embrace change》的建

议，还是建立一个高级的抽象吧！下面我们定义了这样一个接口：

```
public interface DOA {
    public void init()throws Exception ;
    public Student next()throws Exception ;
    public Student previous()throws Exception ;
    public void delete()throws Exception ;
    public void update()throws Exception ;
    public void save()throws Exception ;
}
```

以上是 DOA 接口中主要提的说明：

1. **init** 方法：程序启动时将部分数据加入内存中，等待使用！我们使用包含 **Student** 对象的双向链表。我们放在进入某个具体操作的窗口启动事件中！
2. **save** 方法：程序退出或者在程序退出界面窗口时，将内存中（双向链表）的数据保存到文件中！
3. **next** 方法：和我们查询界面中的向后的按钮对于！即向按钮中就是使用 DOA 中的该方法。**previous** 类似吧！
4. **delete** 方法：和删除按钮的事件关联的 DOA 的方法！
5. **update** 方法：和更新按钮的事件关联的 DOA 的方法！

现在我们程序的大体结构如下：

```
public class Delete extends JPanel implements ActionListener {
    public Delete() {
        initGUI();
    }
    private void initGUI(){}
    public void actionPerformed(ActionEvent e) { }
}
```

我们对文件操作的具体类现在可以创建了！

```
public class FileDOA implements DOA {
    public void init()throws Exception{ }
    public Student next()throws Exception{ }
    public Student previous()throws Exception{ }
    public void delete()throws Exception{ }
    public void update()throws Exception{ }
    public void save()throws Exception { }
}
```

以后有需要可以创建一个 DBDOA 的类，负责从数据库中获取数据。下面是简单的实现！

```
import java.sql.*;
```

```
public class DBDOA implements DOA {
    Connection con ;
    Statement stmt ;
    ResultSet rs ;

    public void init()throws Exception{
```

```

        Class.forName("你数据库驱动的名称");
        String url = "你数据库的 URL";
        String userName = "";
        String password = "";
        con = DriverManager.getConnection(url,userName,password);
        stmt = con.createStatement(int,int ,int);
        rs = stmt.executeQuery("select * from TABLE");
    }

    public Student next()throws Exception{
        Student stu =      ;
        rs.next();
        rs2Sstu(rs,stu);
        return stu ;
    }

    public Student previous()throws Exception{
        Student stu =      ;
        rs.previous();
        rs2Sstu(rs,stu);
        return stu ;
    }

    public void delete()throws Exception{
        rs.deleteRow();
    }

    public void update()throws Exception{
        rs.updateString();
        rs.updateString();
        .....
    }

    public void save()throws Exception {
        rs.close();
        stmt.close();
        con.close();
    }

    private void rs2Sstu(rs,stu){
        //将结果集中的字段一次取出，放入 Student 对象的属性
    }
}

```

其它内容自己填写吧！

不过上面的程序一般要在主要模块启动后启动！一次加入所有的数据，肯定会耗费大量的内存！本博客会有其它文章讲述 ResultSet 的缓存问题！等待吧！

FileDOA 的实现过程中为了避免同样的一次全部加载所有的问题！我们将在下篇给出主要想法！详情见下篇！！

**更多精彩请关注：**

**<http://blog.163.com/miaoxiaodong78/>**