

设计模式实验

一、实验目的

1. 结合实例，熟练绘制设计模式结构图。
2. 结合实例，熟练使用 Java 语言实现设计模式。
3. 通过本实验，理解每一种设计模式的模式动机，掌握模式结构，学习如何使用代码实现这些设计模式。

二、实验要求

1. 结合实例，绘制设计模式的结构图。
2. 使用 Java 语言实现设计模式实例，代码运行正确。

三、实验内容

1. 观察者模式

某文字编辑软件须提供如下功能：在文本编辑窗口中包含一个可编辑文本区和 3 个文本信息统计区，用户可以在可编辑文本区对文本进行编辑操作，第一个文本信息统计区用于显示可编辑文本区中出现的单词总数量和字符总数量，第二个文本信息统计区用于显示可编辑文本区中出现的单词（去重后按照字典序排序），第三个文本信息统计区用于按照出现频次降序显示可编辑文本区中出现的单词以及每个单词出现的次数（例如 `hello :5`）。现采用观察者模式设计该功能，绘制对应的类图并编程模拟实现。

2. 备忘录模式

某文字编辑软件须提供撤销（Undo）和重做／恢复（Redo）功能，并且该软件可支持文档对象的多步撤销和重做。开发人员决定采用备忘录模式来实现该功能，在实现过程中引入栈（Stack）作为数据结构。在实现时，可以将备忘录对象保存在两个栈中，一个栈包含用于实现撤销操作的状态对象，另一个栈包含用于实现重做操作的状态对象。在实现撤销操作时，会弹出撤销栈栈顶对象以获取前一个状态并将其设置给应用程序；同样，在实现重做操作时，会弹出重做栈栈顶对象以获取下一个状态并将其设置给应用程序。绘制对应的类图并编程模拟实现。

3. 状态模式

在某网络管理软件中，TCP 连接（TCP Connection）具有建立（Established）、监听（Listening）、关闭（Closed）等多种状态，在不同的状态下 TCP 连接对象具有不同的行为，连接对象还可以从一个状态转换到另一个状态。当一个连接对象收到其他对象的请求时，它根据自身的当前状态做出不同的反应。现采用状态模式对 TCP 连接进行设计，绘制对应的类图并编程模拟实现。

4. 享元模式

某 OA 系统采用享元模式设计权限控制与管理模块，在该模块中，将与系统功能相对应的业务类设计为享元类并将相应的业务对象存储到享元池中（提示：可使用 Map 实现，key 为业务对象对应的权限编码，value 为业务对象）。用户身份验证成功后，系统通过存储在数据库中的该用户的权限编码集从享元池获取相应的业务对象并构建权限列表，在界面上显示用户所拥有的权限。根据以上描述，绘制对应的类图并编程模拟实现。

四、实验结果

需要提供设计模式实例的结构图（类图）和实现代码。

五、实验小结

请总结本次实验的体会，包括学会了什么、遇到哪些问题、如何解决这些问题以及存在哪些有待改进的地方。