实验7 行为型模式1

1 实验目的

学会用UML设计类图，熟练掌握状态模式。

2 实验环境

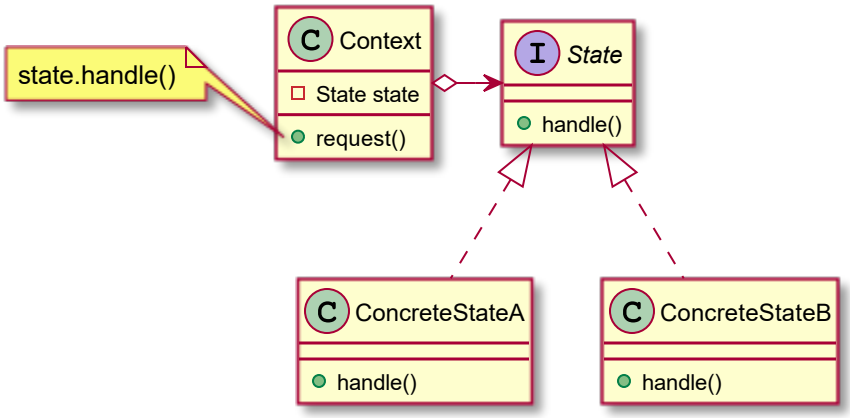
开发环境：JDK 8.0（或更高版本，高版本要下载独立JavaFX）
开发工具：Eclipse
设计工具：StarUML（或PlantUML）

3 基础知识

3.1 状态模式

状态模式允许对象在内部状态发生变化时改变它的行为，对象看起来像修改了类本身。

> Allow an object to alter its behavior when its internal state changes. The object will appear to change its class.



3.2 实现代码

```
public interface IState {
    public void handle(Context context);
}
public class Context {
    IState state;
    public Context(IState state) {
        super();
        this.state = state;
    }
    public void request() {
        state.handle(this);
    }
    public IState getState() {
        return state;
    }
    public void setState(IState state) {
        this.state = state;
    }
}
public class StateA implements IState {
    @Override
    public void handle(Context context) {
        System.out.println("状态A下处理事务");
        context.setState(new StateB());
    }
}
public class StateB implements IState {
    @Override
    public void handle(Context context) {
        System.out.println("状态B下处理事务");
    }
}
public class Test {
    public static void main(String[] args) {
        Context c = new Context(new StateA());
        c.request();
        c.request();
    }
}
```

运行结果：

```
状态A下处理事务
状态B下处理事务
```

4 实验内容

实验1 留言版

问题描述

> 开发一个留言板系统，要求两个基本功能：（1）游客登陆。获取游客登陆信息，例如IP地址。（2）留言功能。登陆状态下可以发送留言消息。（3）退出。登陆状态下输入exit命令退出登陆。运行效果如下：

```
游客登陆（Y/N）：Y
游客[192.168.1.103]登陆成功
*******************************留言板*******************************
************************************************************
沙发
*******************************留言板*******************************
-->沙发        192.168.1.103        Sat Apr 24 20:13:41 CST 2021
************************************************************
exit
*******************************留言板*******************************
-->沙发        192.168.1.103        Sat Apr 24 20:13:41 CST 2021
************************************************************
游客登陆（Y/N）：Y
游客[192.168.1.103]登陆成功
*******************************留言板*******************************
-->沙发        192.168.1.103        Sat Apr 24 20:13:41 CST 2021
************************************************************
板凳
*******************************留言板*******************************
-->沙发        192.168.1.103        Sat Apr 24 20:13:41 CST 2021
-->板凳        192.168.1.103        Sat Apr 24 20:13:51 CST 2021
************************************************************
```

提示与解析：

1、存在登陆和未登陆两种状态，状态接口`ILoginState`用于处理向留言板发送的操作请求。

```
public interface ILoginState {
    public void handle(MessageBoard context);
}
```

2、设计留言板类，包括消息管理和状态管理。

```
public class MessageBoard {
    private List<String> messageList;
    private ILoginState state;
    public static ILoginState LOGIN = new LoginState();
    public static ILoginState LOGOUT = new LogoutState();

    public MessageBoard(ILoginState state) {
        super();
        this.state = state;
        this.messageList = new ArrayList<String>();
    }
    // 客户端对留言板发出请求，留言板根据当前状态运行
    public void request() {
        state.handle(this);
        display();
    }

    public void display() {
        // 显示所有留言
    }
    public void addMessage(String m) {
        // 添加消息
    }
}
```

3、设计具体状态类。

```
public class LoginState implements ILoginState {
    @Override
    public void handle(MessageBoard mb) {
        String str = mb.input();
        if(str.equals("exit")) {
            mb.setState(MessageBoard.LOGOUT);
        }else {
            try {
                String ip = InetAddress.getLocalHost().getHostAddress();
                mb.addMessage(str + "\t" + ip + "\t" + new Date());
            } catch (UnknownHostException e) {
                e.printStackTrace();
            }
        }
    }
}
public class LogoutState implements ILoginState {
    @Override
    public void handle(MessageBoard mb) {
        login(mb);
    }
    private void login(MessageBoard mb) {
        System.out.print("游客登陆 (Y/N) : ");
        if(mb.input().equals("Y")) {
            // 提示游客登陆成功
            // 修改留言板状态
        }
    }
}
```

4、客户端程序采用循环使留言板持续运行。

```
public class Test {
    public static void main(String[] args) {
        MessageBoard mb = new MessageBoard(new LogoutState());
        while(true) {
            mb.request();
        }
    }
}
```

具体要求：

1、调试样例程序，使其正常运行；

2、绘制类图，说明实现过程和使用效果。

5 实验要求

5.2 实验评价

1、完成实验内容（60%）
2、对实验思路进行阐述（20%）
3、对实验过程进行总结（20%）

5.1 实验报告

1、根据要求完成实验内容、思路阐述和总结。
2、截取清晰的核心代码、设计图和效果到报告中；