

Practical Object oriented design (Observer & Mediator)

Solve the following problems.

Problem1: Facebook

In principle a social network service focuses on building online communities of people who share interests and/or activities, or who are interested in exploring the interests and activities of others. Facebook support groups that people can join. Each group has a title, administrative members, a group type (open/ closed), and a list of related groups. If somebody writes on the wall page of the group, the information is broadcasted to all the members and it is visualized in the news feeds of the members. Users should be able to join a group as well as leave a group if they get bored. Once a user has joined a group it will automatically receive any updates that are published on the wall. Which design pattern is the most appropriate to handle this basic functionality of such a Facebook group? Provide the Design as well. The following test code provides details about the group operation:

```
public static void main(String[] args) {  
  
    System.out.println("Testing the Facebook Application");  
  
    //Create a group  
    FacebookGroup dp = new FacebookGroup();  
  
    //Create users  
    FacebookUser user1 = new FacebookUser(dp, "XYZ", 23);  
    FacebookUser user2 = new FacebookUser(dp, "ABC", 20);  
    FacebookUser user3 = new FacebookUser(dp, "AXY", 25);  
  
    //Add users to the newly created group  
    dp.addUser(user1);  
    dp.addUser(user2);  
    dp.addUser(user3);  
  
    //write something on the wall  
    dp.setState("Hello World");  
  
    //users can also write on the wall  
    user1.writeOnTheWall("Hi");  
}
```

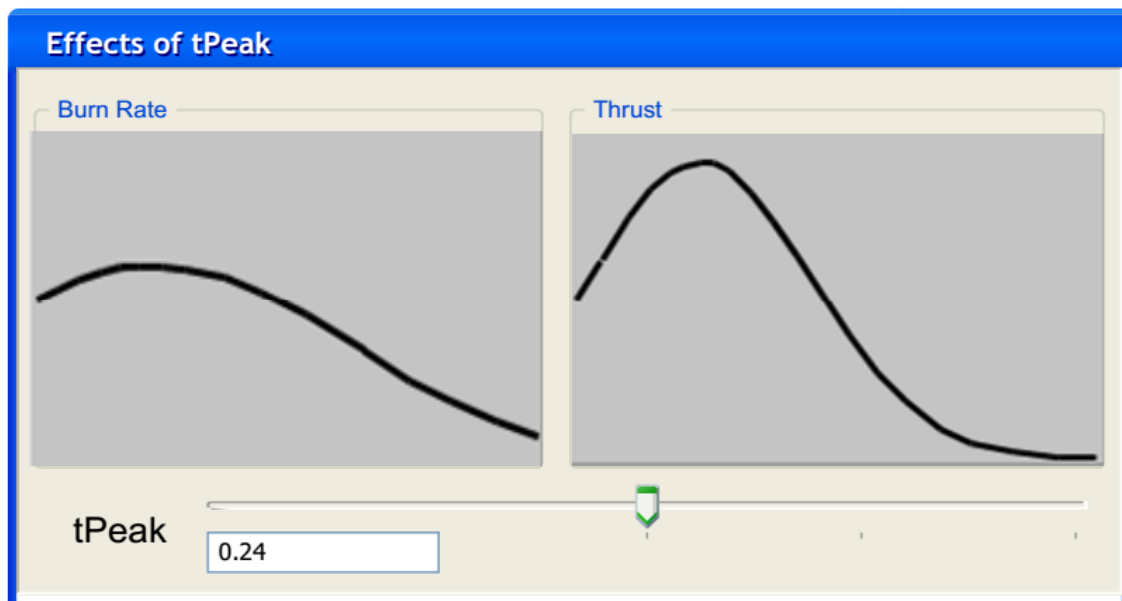
Practical Object oriented design (Observer & Mediator)

Problem2: Groupon

Groupon webpage shows deeply discounted offers from different business located in the city where you are living. As an example, it might show a beauty salon offer that has a 50% discount on women's haircut. You do nothing if you are not interested. If you like the Groupon deal, you select "buy" before the expiration date and your credit card is charged. If enough people select the deal, you will be sent a link for your rebate coupon at the end of the offer. Once you receive your coupon you can claim it at the specified store, usually within a period of time before it expires as noted on the website. If enough people do not join for that deal, the Groupon is cancelled and no one gets it and in this case you will get a refund for your money. In case that you have changed your mind you can also cancel an already made order and you will get a full refund. Provide a design for the specified functionality.

Problem3: Turbo Engine Ignition

You will need now to build for Audi an application (like the one shown in the figure below) that will give the possibility to the turbo ignition engine engineers to experiment visually with parameters that determine the relationships between the car thrust and the surface area of the ignited fuel.



In the new turbo engine that is under development when the engine ignites, the part of its fuel that is exposed to air burns producing thrust. From ignition to maximum burn rate, the burn area increases from the initial ignition area to the full surface area of the fuel.

Practical Object oriented design (Observer & Mediator)

This maximum rate occurs at time t_{Peak} . As fuel burns off, the surface area reduces again until the fuel is consumed. Suppose that the burn rate and thrust equations are:

$$rate = 25^{-(t-t_{peak})^2} \quad thrust = 1.7 \left(\frac{rate}{0.6} \right)^{\frac{1}{0.3}}$$

The application from the figure above shows how t_{Peak} affects the burn rate and thrust of a car. As a user moves the slider, the value of t_{Peak} changes, and the curves take on new shapes. The value for t_{Peak} can be also entered numerically in the edit box below the slider. Design the application for above requirements?

Problem4: War games

Imagine a war zone where armed units are moving into the enemy's territory. Armed units can include soldier, tank, grenadier, and sniper units. The strategy being employed is that whenever one unit attacks, other units should stop attacking and take cover. To do so, the unit that is currently attacking needs to notify the other units. Provide object oriented design to implement the above functionality.