

# Exercise 5

15.11.2001

**Due date: 28.11.2002**

---

## Assignment 5.1

This assignment introduces the facade design pattern. The facade pattern provides a unified interface to a set of interfaces in a subsystem. A facade defines a higher-level interface that makes the subsystems easier to use.

Implement a "credit institute application" that tests if a customer can be trusted. That means if the credit institute's policy allows giving the customer a credit. The application provides a simple interface to a "larger" subsystem of classes. The system consists of

1. Credit institute application that delegates requests to the appropriate subsystem
2. Subsystems
  - 2.1 Subsystem Bank (can tell about the customer's accounts )
  - 2.2 Subsystem Credit Policy (defines the credit policy, i.e. says how many saving a customer should have on his saving account to give a credit of x Euro.)
  - 2.3.Subsystem PoliceDepartment (can tell if the customer is trustworthy)

The credit institute gives away a credit, if the ploice department has no file of the customer and the customer has enough savings on his account for the request of a credit of xxx Euro.

## Assignment 5.2

This assignment introduces the mediator design pattern. The mediator design pattern defines an object that encapsulates how a set of objects interact. It promotes loose coupling by keeping objects from referring to each other explicitly, and it lets you vary their interaction independently.

Implement a "chat-room application".

1. The meditor class is the chat-room interface.
2. The concrete mediator is our simple chatroom. This class implements the coordination between the participants and executes the rules of simple and VIP participants. It knows and maintains the list of active participants. Participants have to register with the chatroom application.
3. Participants.
  - 3.1 Simple participant (the simple participant can only send 2 message per hour or so...)
  - 3.2 VIP participant (can write an unlimited number of messages)

How to do it. Do it simple (do not implement a full-scale internet enabled application, however you are welcome to do it anyway...): Implement the application in a single project. Messages are sent by specifying the addressee, sender, the message itself and sending it to the chatroom application. The chatroom sends the messages to the addressee (or a set of addressee). When a new participant is registering the chatroom sends a message to all participants that a new participant is online.

For ambitious students (everyone of you is ambitious, aren't you ;-)!): Implement the participant such that each of them runs in its own thread and has a user interface to submit messages.

---