Assignment 10

16.05.2006 **Due date: 22.06.2006**

Exercise 1

Reengineer your implementation of the library, will all features we added during the last assignments (CocoR and transformations).

In the Proseminar we discussed design issues of an architecture of the library application that promote modularization leading to extensibility, modifiability, adaptability etc.

When redesigning your implementation, think about a good **ratio of coupling and cohesion** of the objects that you use in your application. Consider **architectural patterns** and if you could use **design patterns**. Also, consider the following **principles** that you apply when doing an OO-design:

- Liskov Substitution Principle (http://www.objectmentor.com/resources/articles/lsp.pdf) - Inheritance should ensure that any property proved about supertype objects also holds for subtype objects.
- Open-Closed Principle (http://www.objectmentor.com/resources/articles/ocp.pdf) - Software entities should be open for extension, but closed for modification.
- Dependency Inversion Principle(http://www.objectmentor.com/resources/articles/dip.pdf) - High-level modules should not depend on low-level modules. Both should depend on abstractions.
- Interface Segregation Principle(http://www.objectmentor.com/resources/articles/isp.pdf) - Clients should not be forced to depend upon interfaces that they do not use.

When presenting your application at the Proseminar **FOCUS ON DESIGN** issues.

- Explain the design pattern that you have used.
- Be able to explain why your application benefits from using these design patterns.
- Which principle do the design patterns implement and why?
- Be able to present a UML class-diagram (static part) and a sequence diagram (dynamic part) of your application.
- Did you consider the Model-View-Controller pattern ?

Exercise 2

After having impemented and refactored the library application until now, we are ready to analyse how good the implementation is ! - How do we do it ?

Download JDepend (http://www.clarkware.com/software/JDepend.html), run JDepend on your application, and analyse the results.

You should be able to explain:

- what JDepend is
- why you may want to use it
- what kind of design quality metrics JDepend calculates
- each design quality metric in detail
- the results for your application !
- What to refactor, reengineer and redesign in order to improve the design.
