Prof. Dr. Sebastian Bremm
Prof. Dr. Jürgen Jung
Prof. Dr. Peter Thoma

Sheet 1

Architecture and classes

Library Management

The university's library needs a new library software (LibSoft) for managing books and rentals. You are a team of students and tasked to develop this software as part of a student project. Before starting coding, you will have to come up with the structure of LibSoft and a draft class diagram.

Exercise 1

You will need an architecture for planning your software development project and for fostering a modular design. Create an architecture diagram that encompasses at least the following modules:

- *lender module*: managing all users that can lend books from the library
- *book inventory*: a collection of all books that can be borrowed from the library
- *rental management*: functionality for lending and returning books
- *administration console*: any functionality requiored for admninistrating the library software

The architecture diagram needs to fulfill the following requirements:

1. A concise **description** is required for each module, explaining its purpose and functionality it encompasses.
2. Assign **links** (relationships) between modules where applicaple.
3. Don't forget to provide a **legend** explaining all symbols used in the diagram.

Exercise 2

After defining the overall structure of LibSoft by an architecture diagram, you now need to consider the classes for implementing the system. Create a UML class diagram as an initial draft for further design activities and, later on, developing the software.

The class diagram needs to include the following classes:

Title Representing bibliographical information on books

Copy an individual instance of a book that can be borrowed from the library

Rental representing the rental of a book copy for a specific lender and time

Lender a person renting a copy (of a book)

Reservation a lender wants to borrow a book that is currently borrowed by somebody else

Each class needs to consist of the following:

1. A concise **description** explaining the purpose of a class as well as the real world entity it is representing.
2. Classes need to provide attributes with type information. Some may have more, other less attributes. Your classes should have in **average three** attributes.
3. Any relevant **association** needs to be visible in the class diagram.
4. Provide methods where appropriate. No get- and set-methods are required!

Exercise 3

Each class from exercise 2 needs to be located within one module from exercise 1. Assign each

of your classes to the respective module in the arcitecture. You can use a table consisting of two columns having class names in the left column and the corresponding module names in the column on the right hand side.

Exercise 4

Create source code in an object-oriented programming language (e.g. Java or C++) based on your classes in the class diagram. Each class needs to be specified in program code including attrbutes and methods.

Expected Deliverables:

You are expected to submit a single PDF file containing your solution. It must contain the following diagrams:

1. *Architecture diagram (UML or any other notation)*
2. *UML class diagram*
3. *Table assigning classes to modules*
4. *Source code*

.