KTH Microelectronics and Information Technology

Exam in ID2207 Modern Methods in Software Engineering, 2007-10-26, 15:00-20:00

Rules

This exam is "closed book" and you are not allowed to bring any material or equipment (such as laptops, PDAs, or mobile phones) with you. The only exceptions are English to "your favorite language" dictionary and pencils.

Instructions

- Please read the entire exam first!
- Write clearly
- Each sheet of paper must contain your name, "personnummer", Problem number and a unique sheet number
- Write only on one page of a sheet. Do not use the back side
- Only one Problem must be reported on each sheet
- If more than one sheet is needed the continuation should be clearly noted on the beginning of each sheet and the sheet numbers used should be consecutive
- Always motivate your answers. Lack of clearly stated motivation can lead to a reduction in the number of points given
- The tasks are not necessarily sorted in order of difficulty. If you get stuck it might be a good idea to go on to the next task.

Grading

The grades depend on the sum of exam and bonus points n:
$n < 50$ fail (F)
$50 \leq n < 60$ grade E
$60 \leq n < 70$ grade D
$70 \leq n < 80$ grade C
$80 \leq n < 90$ grade B
$90 \leq n$ grade A

GOOD LUCK!

Mihhail Matskin, mobile 0704614269

Problem I. General questions

a) What is a Principle of Falsification in modeling? Why and how it is applied in software design?

(5p)

b) Which types of decomposition do you know? Briefly explain and give their advantages and disadvantages.

(4p)

Problem II. Software Life Cycle

a) Unified Software Development Process (UP) considers phases, iterations and workflows in software life cycle. Explain how they are related.

(5p)

b) Specify which of the following decisions were made during requirements or system design:

• "The ticket distributor is composed of a user interface subsystem, a subsystem for computing tariff, and a network subsystem managing communication with the central computer."

• "The ticket distributor will use PowerPC processor chips."

• "The ticket distributor provides the traveler with an on-line help."

(4p)

Problem III. UML and OOP

a) Using only **one class** draw a class diagram which states that a node in a network structure has successor(s) and predecessor(s) nodes and if two nodes are in the same region then they are mutually exclusive.

(5p)

b) For textual description of use cases we use a template composed of six fields.

- Explain what are these fields?

(4p)

- How do we represent include and extend relationships in the textual description of use cases?

(6p)

Problem IV. Requirements Elicitation

a) What are main types of scenarios? Briefly explain them.

(5p)

b) What are main types of requirements? Briefly explain each of them

(4p)

Problem V. Requirements Analysis

a) There are different approaches to classes identification. In practice, the process of class discovery is likely to be guided by different approaches at different times. Give a scenario of a mixed approach to class discovery involving at least 4 other approaches.

(6p)

b) Analysis model is composed of some other models. What are these models and when each of them is dominant?

(5p)

Problem VI. System Design

a) What are benefits of layered architecture? What are main problems with this architecture?

(4p)

b) Older compilers were designed according to a pipe and filter architecture, in which each stage would transform its input into an intermediate representation passed to the next stage. Modern development environments, including compilers integrated into interactive development environments with syntactical text editors and source-level debuggers, use a repository architecture. Identify the design goals that may have triggered the shift from pipe and filter to repository architecture.

(5p)

Problem VII. Object Design - Reuse

a) Indicate which occurrences of the following inheritance relationships are specification inheritance and which are implementation inheritance:

- A Rectangle class inherits from a Polygon class.
- A Set class inherits from a BinaryTree class.
- A Set class inherits from a Bag class (a Bag is defined as an unordered collection).
- A Player class inherits from a User class.
- A Window class inherits from a Polygon class.

(5p)

b) What is Observer pattern? Where it is applicable? Give an example

(6p)

Problem VIII. Object Design – Interface design

a) How does contract inheritance work? Explain for each type of constrains.

(5p)

b) Assume that we have a Rectangle class that inherits from a Parallelogram class. Write invariant(s) in OCL for the Rectangle class.

(5p)

Problem IX. Moving to Code

a) Explain the following mapping concepts: model transformation, forward engineering, reverse engineering and refactoring. Draw a figure showing their relations to model space and source code space.

(5p)

b) Explain different ways of mapping inheritance into relational database schema. Give their pros and cons.

(6p)

Problem X. Testing.

a) Explain modified sandwich testing strategy. What are advantage(s) and disadvantage(s) of this strategy?

(6p)

----------End of Exam----------