**KTH Microelectronics
and Information Technology**

Exam in 2G1522 Modern Methods in Software Engineering, 2005-05-25, 14:00-19:00

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 an 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 (U)
$50 \leq n < 67$ grade 3
$67 \leq n < 84$ grade 4
$84 \leq n$ grade 5

GOOD LUCK!

Mihhail Matskin, mobile 0704614269

Problem I. General questions

a) Explain dual role of software.

(4p)

b) What are basic ways to deal with complexity?

(4p)

c) In the following description, explain when the term “account” is used as an application domain concept and when as a solution domain concept:

“Assume you are developing an online system for managing bank accounts for mobile customers. A major design issue is how to provide access to the accounts when the customer cannot establish an online connection. One proposal is that accounts are made available on the mobile computer, even if the server is not up. In this case, the accounts show the amounts from the last connected session.”

(4p)

Problem II. Software Life Cycle

a) What is difference between activity-centered and entity-centered life cycles? Which of them is dominating now?

(4p)

b) Draw Boehm’s spiral model as a UML activity diagram. Does the activity diagram improve readability compare to the original figure?

(5p)

Problem III. UML and OOP

a) What is difference between an actor, a class and an instance? Give examples.

(4p)

b) Draw a class diagram representing the relationship between parents and children. Take into account that a person can have both a parent and a child. **Requirement: your class diagram must contain only one class**

(7p)

Problem IV. Requirements Elicitation

a) Below are examples of nonfunctional requirements. Specify which of these requirements are verifiable and which are not.

- "The system must be usable."
- "The system must provide visual feedback to the user within 1 second of issuing a command."
- "The availability of the system must be above 95%."
- "The user interface of the new system should be similar enough to the old system such that users familiar with the old system can be easily trained to use the new system."

(4p)

Problem V. Requirements Analysis

a) What are components of the Analysis model and which UML diagrams are used for their presentation?

(4p)

b) Consider a file system with a graphical user interface, such as Macintosh's Finder, Microsoft's Windows Explorer, or Linux's KDE. The following objects were identified from a use case describing how to copy a file from a floppy disk to a hard disk: File, Icon, TrashCan, Folder, Disk, Pointer. Specify which are entity objects, which are boundary objects, and which are control objects.

(5p)

Problem VI. System Design

a) Explain notions of coupling and cohesion. What are goals of system decomposition in terms of coupling and cohesion?

(4p)

b) In many architectures, such as the three- and four-tier architectures, the storage of persistent objects is handled by a dedicated layer. In your opinion, which design goals have lead to this decision?

(5p)

Problem VII. Object Design - Reuse

a) What is basic idea of the Liskov Substitution principle? Which type of inheritance it defines?

(6p)

e) Explain Adapter pattern. What is difference between class Adapter pattern and object Adapter pattern?

(6p)

Problem VIII. Object Design – Interface design

a) Which different roles play developers in Object design and how it is reflected in UML diagrams?

(4p)

b) Consider the List interface in the java.util package for ordered collections of objects. Write preconditions and post conditions in OCL for the following operations:

- **`int size()`** returns the number of elements in the list.
- **`void add(Object e)`** adds an object at the end of the list.
- **`void remove(Object e)`** removes an object from the end of the list.
- **`boolean contains(Object e)`** returns true if the object is contained in the list.
- **`Object get(int idx)`** returns the object located at index idx, 0 being the index of the first object in the list.

(10p)

Problem IX. Moving to Code

a) Explain how a unidirectional, one-to-one association can be realized in source code. Give and example

(6p)

Problem IX. Extreme Programming.

a) What are main ideas of Manifesto of the Agile Software Development?

(4p)

Problem X. Testing.

a) What are guidelines for selecting equivalence classes in testing?

(5p)

b) What is path testing? Is it white-box or black-box technique? Does it allow to identify all faults? Explain your answer.

(5p)

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