KTH Microelectronics and Information Technology

Exam in ID2207 Modern Methods in Software Engineering, 2008-10-20, 09:00-13:00

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
- Sort your sheets according to the problem’s numbering
- 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) A programming language is a notation for representing algorithms and data structures. List two advantages and two disadvantages of using a programming language as sole notation throughout the development process.

(5p)

b) A passenger aircraft is composed of several millions of individual parts and requires thousands of persons to assemble. A four-lane highway bridge is another example of complexity. The first version of Word for Windows, a word processor released by Microsoft in November 1989, required 55 person-years, resulted into 249,000 lines of source code, and was delivered 4 years late. Aircraft and highway bridges are usually delivered on time and below budget, whereas software is often not. Discuss what are, in your opinion, the differences between developing an aircraft, a bridge, and a word processor, which would cause this situation.

(5p)

Problem II. Software Life Cycle

a) Give an explanation of Spiral Model? What are main problems of waterfall model that Spiral model overcomes?

(5p)

b) Software life-cycles may be different for different projects. Why?

(4p)

Problem III. UML and OOP

a) Draw a UML activity diagram describing the dependency between activities for a life cycle in which requirements, design, implementation and testing occur both concurrently and iteratively.

(5p)

b) What is/are difference(s) between Sequence diagrams and Collaboration diagrams? When each of them is more appropriate to use?

(4p)

c) Both Activity and Statechart diagrams are described as graphs. What is difference between Activity and Statechart graphs?

(4p)

Problem IV. Requirements Elicitation

a) Maintaining traceability during requirements and subsequent activities is expensive, because of the addition information that must be captured and maintained. What are the benefits of traceability that outweigh this overhead?

(4p)

b) The need for developing a complete specification may encourage an analyst to write detailed and lengthy documents. Which requirement validation criteria(s) may encourage an analyst to keep the specification short?

(4p)

c) Name categories of nonfunctional requirements suggested by FURPS+ model. Which of them are quality requirements and which of them are constraints or pseudo requirements?

(5p)

Problem V. Requirements Analysis

a) Consider a file system with a graphical user interface, such as Microsoft's Windows Explorer. 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.

(4p)

b) Briefly explain main idea of Aspect-Oriented Programming.

(4p)

c) Explain the approaches to class identification?

(5p)

Problem VI. System Design

a) Explain how the results from the Requirements Analysis can be used in different System Design activities.

(5p)

b) Explain Repository architectural style.

(5p)

Problem VII. Object Design - Reuse

a) Explain difference between implementation inheritance, specification inheritance and delegation. How they all are shown in UML diagrams?

(5p)

b) Explain Command pattern. When is it applicable?

(6p)

Problem VIII. Object Design – Interface design

a) Consider a sorted binary tree data structure for storing integers. Write invariants in OCL denoting that

• All nodes in the left subtree of any node contain integers that are less than or equal to the current node, or the subtree is empty.

• All nodes in the right subtree of any node contain integers that are greater than the current tree, or the subtree is empty.

(6p)

Problem IX. Moving to Code

a) Explain realization of a bidirectional, many-to-many association in source code.

(5p)

Problem X. Testing.

a) Briefly explain which different types of tests are performed during testing. Who performs each of them?

(5p)

Problem XI. Agile Software Development.

a) What is test-driven (test-first) software development? Briefly explain.

(5p)

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