KTH Microelectronics
and Information Technology

Exam in ID2207 Modern Methods in Software Engineering, 2014-10-29, 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

If n is amount of your exam points and m is your bonus points earned in autumn 2014 then:

$n+m < 50$ fail (F)
$50 \leq n+m < 60$ grade E
$60 \leq n+m < 70$ grade D
$70 \leq n+m < 80$ grade C
$80 \leq n+m < 90$ grade B
$90 \leq n+m$ grade A

GOOD LUCK!

Mihhail Matskin, mobile 0704614269

Problem I. General questions

a) What is difference between the problem domain and the solution domain?

(3p)

b) 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)

c) 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.

(4p)

Problem II. Software Life Cycle

a) For each of the following documents, indicate in which phase(s) of the software life-cycle it is produced: final user manual, architectural design, module specification, source code, test plan, preliminary user manual, detailed design, project plan, cost estimate, test report, documentation.

(5p)

b) Draw a UML class diagram representing Unified Software Development Process (UP)?

(5p)

c) We considered 3 formal software development models: inductive, deductive and transformational. Briefly (without technical details) explain main features of each of these models

(5p)

Problem III. UML and OOP

a) Classify the following relationships as either and inheritance relationship, an aggregation relationship or a general association:

Car - Lincoln town car

Person – student

Library – Library employee

Book – Copy

Car – Driver

Library employee – Book loan

Class – Students

(5p)

b) Using only one class draw a class diagram which states that a course has successor(s) and prerequisite(s) and if two courses cover nearly the same material, taking one of them may preclude a student from taking the other and vice versa.

(6p)

Problem IV. Requirements Elicitation

a) We identified the following types of scenario: As-Is, Visionary, Evaluation and Training scenario. Briefly explain each of them.

(4p)

b) 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)

Problem V. Requirements Analysis

a) There are trade-offs between different design goals. Give at least 3 examples of such trade-offs and briefly explain them.

(4p)

b) 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 of the identified objects are entity, boundary and control objects.

(4p)

c) Which restrictions of Object-oriented programming can be resolved in Aspect oriented programming?

(5p)

Problem VI. System Design

a) Explain the Pipe and Filter architectural style. For which systems it is suitable and for which systems it is not suitable?

(5p)

b) Explain the following terms: coupling, cohesion, partitioning and layering.

(4p)

Problem VII. Object Design - Reuse

a) What are pros and cons for inheritance and delegation?

(4p)

b) Explain Command pattern. When is it applicable?

(5p)

Problem VIII. Object Design – Interface design

a) Which types of navigations through a class diagram exists? Give examples of constraints using each type of navigation

(5p)

b) 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 two ways of mapping inheritance to relational database. When each of them is more preferable?

(5p)

Problem X. Testing.

a) If a program has 2 integer inputs and each can be 32-bit integer, how many possible inputs does this program have?

(4p)

b) Why do we use equivalence classes in testing?

(4p)

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