



KTH Microelectronics  
and Information Technology

## **Exam in ID2207 Modern Methods in Software Engineering, 2018-10-25, 08:00-12:00**

### 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 amount of your bonus points earned in autumn 2018 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 are basic ways to deal with complexity?

(3p)

b) Which of the following are models?

- a UML class diagram
- a set of UML class diagrams describing the classes in a software system
- a 1:100 scale clay replica of a new sports car that will be used to test its aerodynamics in a wind tunnel
- a full-scale, working prototype of a new sports car

(3p)

c) Why do we apply principle of Falsification in software engineering?

(5p)

## Problem II. Software Life Cycle

a) Give an explanation of V-Model? What are/is its main difference(s) compare to Waterfall model?

(5p)

b) Explain what is common and what is different between transformational and deductive synthesis. Explain what is common and what is different between transformational synthesis and Model Driven Architecture (MDA)

(6p)

## Problem III. UML and OOP

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

(3p)

b) Consider the object model in Figure 1. Given your knowledge of the Gregorian calendar, modify the model such that a developer unfamiliar with the Gregorian calendar could deduce the number of days in each month, the number of weeks in each month, the number of months in a year and the number of days in a week? Identify additional classes and associations if necessary.

(6p)

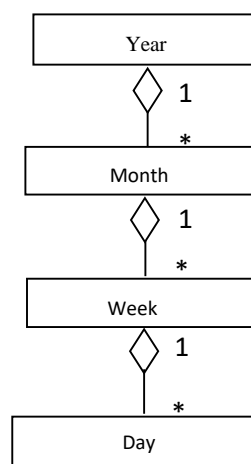


Figure 1

c) Which of the following kinds of use-case(s) is (are) always independently meaningful and operational?

Explain you answer

- 1) An extension(extending) use-case
- 2) An extended use-case
- 3) An including(base) use-case
- 4) An included use-case

(4p)

d) Based on Activity Diagram in Figure 2, which of the following statements are true?

- 1) The 'Insert NewsItem' activity is executed many times
- 2) The 'Rollback Changes' and 'Invalidate Session,, activities are taken in parallel.
- 3) The 'Rollback Changes' and 'Invalidate Session,, activities are executed many times.
- 4) The 'Rollback Changes' and 'Generate Confirmation,, activities are taken in parallel
- 5) If session is valid then confirmation is generated.

(4p)

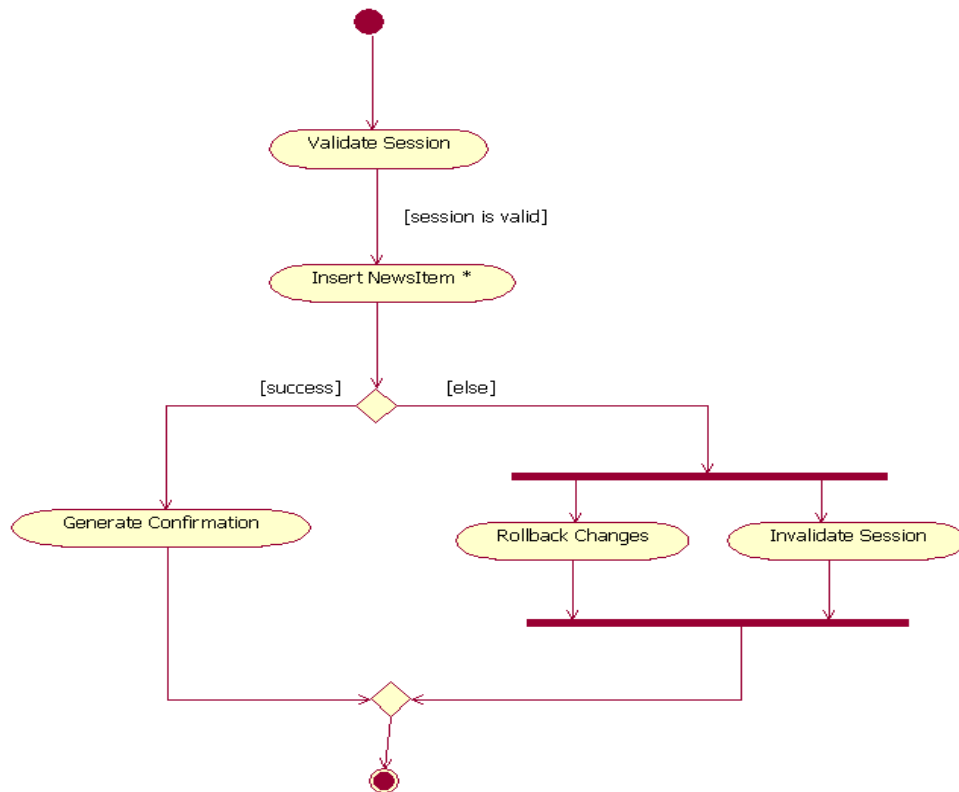


Figure 2

e) Finish the correct statements about Use Cases relationships: *Use ...when you are repeating yourself in two or more separate use cases and you want to avoid repetition. Use ..... when you have one use case that is similar to another use case but does a bit more. Use ..... when you are describing a variation on normal behavior and you wish to use a controlled form, declaring your extension points in your base use case.*

- 1) include
- 2) generalization
- 3) extend
- 4) delegation

(4p)

## Problem IV. Requirements Elicitation

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

(3p)

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 different approaches to class 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.

(4p)

b) Give an example of a problem where Aspect-Oriented Programming is useful to apply. Your example must be different from the examples presented in the Lecture Notes of the course.

(5p)

## Problem VI. System Design

a) Assume that we classified design goals into five categories: performance, dependability, cost, maintenance, and end user. Assign one or more categories to each of the following goals:

1. Users must be given a feedback within 1 second after they issue any command.
2. The TicketDistributor must be able to issue train tickets, even in the event of a network failure.
3. The housing of the TicketDistributor must allow for new buttons to be installed in the event the number of different fares increases.
4. The AutomatedTellerMachine must withstand dictionary attacks (i.e., users attempting to discover an identification number by systematic trial).
5. The user interface of the system should prevent users from issuing commands in the wrong order.

(4p)

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

(4p)

c) 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 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) Consider an application that must select dynamically an encryption algorithm based on security requirements and computing time constraints. Which design pattern would you select? Draw a UML class diagram depicting the classes in the pattern and justify your choice.

(5p)

b) Explain how delegation, implementation and specification inheritance are shown in UML class diagrams.

(3p)

### Problem VIII. Object Design – Interface design

a) Consider a simple intersection with two crossing roads and four traffic lights. Assume a simple algorithm for switching lights, so that the traffic on one road can proceed while the traffic on the other road is stopped. Model each traffic light as an instance of a TrafficLight class with a state attribute that can be either red, yellow, or green. Write invariants in OCL on the state attribute of the TrafficLight class that guarantee that the traffic cannot proceed on both roads simultaneously. Add associations to the model to navigate the system, if necessary.

(6p)

### 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.

(3p)

b) Explain realization of a unidirectional, one-to-one association in source code.

(4p)

### Problem X. Testing.

a) Explain the following testing concepts: Reliability, Fault, Erroneous state, Failure.

(4p)

### Problem XI. Agile Software Development.

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

(3p)

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