EXAM 2024

Chapters in the text-book that are less relevant to the exam

- Chapter 3 – Project organization and Communication
- Chapter 12 – Rational Management
- Chapter 13 – Configuration Management
- Chapter 14 – Project Management

Grading

- In order to pass exam (grade E) you have to get at least 36 points from questions from Part I (this includes your bonus points).
- In order to get grade D you have to pass exam (see item 1) and get 49 points from all questions (this includes your bonus points).
- In order to get grade C you have to pass exam (see item 1) and get 63 points from all questions (this includes your bonus points).
- In order to get grade B you have to pass exam (see item 1) and get 74 points from all questions (this includes your bonus points).
- In order to get grade A you have to pass exam (see item 1) and get 85 points from all questions (this includes your bonus points).

Part I.
General questions (48p)

1) **Which types of decomposition do you know? Briefly explain and give their advantages and disadvantages. (4p)**

2) **A programming language is a notation for representing algorithms and data structures. List two advantages and two disadvantages of using a programming language as the only notation throughout the development process (4p)**

Software Life-Cycle

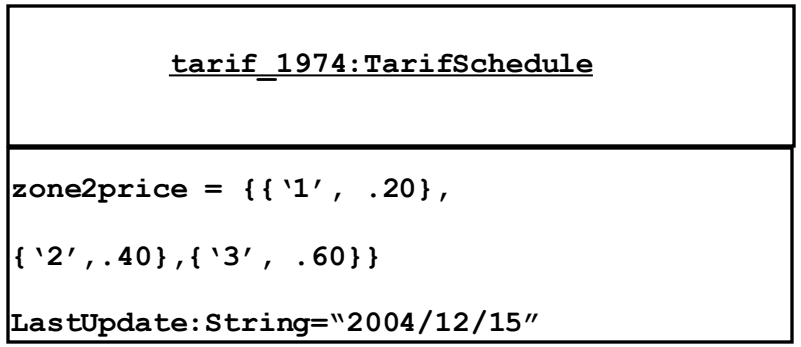
3) Specify which of the following decisions were made during requirements or system design (explain your answer):

- *"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"*

(3p)

Modelling with UML

4) Give an example of object description in UML class diagram notation.



Requirements Elicitation

5) Specify which of these requirements are verifiable and which are not. Explain your answer.

(4p)

- *"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."*

Requirements Analysis

6) Both sequence diagrams and statecharts diagrams can be used in analysis. When each of them is more appropriate?

(4p)

7) What is difference between fork and stair structure of UML Sequence diagram? For which types of system each of them is more suitable?

(4p)

System Design

8) What are advantages and disadvantages of open and closed layered architectures? Which of them is more suitable in which cases?

- Closed architectures are more portable.
- Open architectures are more efficient

(4p)

Object Design

9) What are pros and cons for inheritance and delegation

(4p)

10) What is Observer pattern? Where it is applicable? Give an example using UML diagrams

(5p)

11) Explain meaning of invariant, pre-condition and post-condition. Give their examples

(4p)

Testing

12) What are pro and cons for black-box and white-box testing?

(4p)

Part II (25p)

13) Explain inductive program synthesis. What is/are its advantage(s) and what is/are its drawback(s)?

(5p)

14) Explain difference between <<extend>> and <<include>> in use case diagrams.

(5p)

15) 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 prevent a student from taking the other and vice versa.

(6p)

Part II

16) In the course, 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 and briefly explain your answer:

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

(5p)

Part II

17) Consider the following design goals. For each of them, indicate the candidate pattern(s) you would consider to satisfy each goal and briefly explain your answer:

(4p)

- **Given a legacy banking application, encapsulate the existing business logic component.**
- **Given a chess program, enable future developers to substitute the planning algorithm that decides on the next move with a better one.**
- **Given a chess program, enable a monitoring component to switch planning algorithms at runtime, based on the opposing player's style and response time.**
- **Given a simulation of a mouse solving a maze, enable the path evaluation component to evaluate different paths independently of the types of moves considered by the mouse.**

Part III (24p)

18) Explain the idea of Aspect-oriented programming. Give example(s)?

(6p)

19) Organize the following set of classes into hierarchies: Circle, Point, Rectangle, matrix, Ellipse, Line and Plane. You can add any additional classes into your hierarchy.

(6p)

Part III. (24p)

20) Consider a traffic light system at a four-way crossroads (e.g., two roads intersecting at right angles). Assume the simplest algorithm for cycling through the lights (e.g., all traffic on one road is allowed to go through the crossroad while the other traffic is stopped). Identify the states of this system and draw a statechart describing them. Remember that each individual traffic light has three states (i.e. green, yellow, and red).

(7p)

Part III

20) 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.**