

**1)** UML modeling language is based on a *metamodeling* approach. UML specification defines the concepts/elements of the language at different levels of abstraction. (6p)

a) How many and what layers does this *metamodeling* approach have? At which of these layers situates the *UML – metamodel* language itself and at which of these layers situates the user defined application model?

b) Inside one layer new elements/concepts are defined as specializations of the more general elements (using *subClassOf* relationship). What is the relationship between elements at different *metamodeling* layers (i.e. relationship over the layer border)?

**2)** UML provides means to model *concurrency* at several levels in a model. *Parallelism* is a carrying principle of the UML modeling semantics meaning that all behavior may in principle occur simultaneously if not explicitly defined otherwise. How can you explicitly define or constrain a concurrent behavior of an object in your model? Consider different levels of behavior ( class or object level, state-machine level, operation level ).(6p)

**3)** Requirements specification (vaatimusten määrittely) is an essential starting phase of the software and system development process. A requirements specification document (vaatimusdokumentti) will be created as an outcome of this phase. The following list contains a few concepts related to requirement management and specification document. Explain these concepts and their meaning.

a) functional (toiminnalliset) and non-functional (ei-toiminnalliset) requirements (2p)

b) user requirements (käyttäjävaatimukset) and system requirements (Systeemi-vaatimukset) (2p)

c) Scenario (Skenaario)(2p)

**4)** Structural modeling question

a) The development of embedded software is more challenging than PC-software. Describe briefly 2 such challenges (2p)

b) To what extent does UML2 structural modeling help address these challenges that you mentioned in part a)? (2p)

**5)**

a) Describe with a few sentences what does the concept, *Design Pattern*, mean in the object oriented programming context. What are the most important parts of the DP description? (3p)

b) Describe the idea, structure and functioning of EITHER the *Monitor-Actuator Pattern* OR the *Observer Pattern* (a.k.a Publish Subscribe pattern). Draw also some kind of diagram of the collaborating objects or classes for support of your explanation. (3p)