# Valdkonna- ja arhitektuurimustrid

Ülesannet (kokku kuni 16 punkti) võib lahendada grupis, kus on kuni kolm inimest. Teema peab olema unikaalne ja see tuleks kooskõlastada (postitada Moodle vastavasse foorumisse). Teema registreerimise tähtaeg on 8. detsember.

Mõned võimalikud teemad:

1. variant. Kirjeldada valdkonnamustrit:

1.1.Raha

1.2. Füüsikaline suurus *(Quantity)*

1.3. Osapool (inimene ja organisatsioon)  *(Party)*

1.4. Toode

1.5. Arve

1.6. Inventari raamat

1.7. Protsess

1.8. Reegel

1.9. Olekumasin, selle metamudel (mitte GoF olek)

1.10. Seose mudel

…

2. variant. Kirjeldada arhitektuurimustrit:

2.1. Kolmekihiline arhitektuur

2.2. R.C. Martini *Clean Architecture*

2.3. *Object-Relational Mapping* (ORM)

2.4. Mikroteenused (*Microservices*)

2.5. Relatsioonilise AB ja OO kihi seos: *Data Mapper*

2.6. Relatsioonilise AB ja OO kihi seos: *Unit of Work*

2.7. Relatsioonilise AB ja OO kihi seos. Seoste modelleerimine: *Identity Field, Foreign Key mapping, Association table mapping, Dependant mapping* (koos).

2.8. Relatsioonilise AB ja OO kihi seos alamklasside kirjeldamisel: *Single Table Inheritance*

2.9. Relatsioonilise AB ja OO kihi seos alamklasside kirjeldamisel: *Class Table Inheritance*

2.10. Relatsioonilise AB ja OO kihi seos alamklasside kirjeldamisel: *Concrete Table Inheritance*

2.11. Relatsioonilise AB ja OO kihi seos alamklasside kirjeldamisel:  *Inheritance Mappers*

2.12. Relatsioonilise AB ja OO kihi seos keerukate objektide haldamisel andmebaasis: *Embedded Value, Serialized LOB*

2.13. Relatsioonilise AB ja OO kihi seos, paindlik lähenemine: *Metadata Mapping*

2.14. Relatsioonilise AB ja OO kihi seos mingi konkreetse püsivusraamistiku (*persistence framework*) abil: Pythoni SQLAlchemy või Java Hibernate vms.

2.15. *Docker* tarkvaraarenduses

2.16. *Model View Controller (MVC*)

…

3. GoF ja testimise mustrid

3.1.. Mingi ülesannetega katmata keerukam GoF muster: *Abstract Factory, Builder, Adapter, Decorator, Flyweight, Proxy, Chain of responsibility, Command, Interpreter, Mediator, Memento, Visitor* (üks neist)*.*

3.2. *Mock object*  testimises

3.3. Testandmeid genereerivad raamistikud: *Hypothesis* või *QuickCheck* vms.

…

4. Võib pakkuda enda teema. Võimalikud mustrite allikad:

M. Fowler. *Patterns of Enterprise Application Architecture.*

M. Fowler. *Analysis Patterns.*

E. Evans. *Domain-Driven Design.*

R. C. Martin*, Agile Software Development*

R. C. Martin*, Clean Architecture*

*…*

Mis siis see kodutöö peab sisaldama selleks, et saada 16 punkti:

1. Mudel (CASE vahendis) ja selle selgitus (ca 2 lk)... loetav tekst (kuni 6 punkti)
2. Kood ja testid. Valdkonnamustri korral esitada täielik kood. Arhitektuurimustri korral pisike funktsionaalsus, mis läbib kõiki kihte ja allsüsteeme (*architectural spike*). Mittestandardsete välissüsteemide korral tuleks kasutada mock (asendus)- objekte (mälus hoitavad/ *in-memory* andmebaasid jms) . (kuni 6 punkti)
3. Ettekanne (lisaülesanne) (kuni 4 punkti)

Ka ilma ettekandeta tuleks tööd siiski õppejõule praktikumis esitleda, mitte ainult Moodlesse üles laadida.

Kuidas tekst vormistada?

Nagu kirjalik töö ikka

1. Sissejuhatus (umbes 10% tööst, kus on sees probleem ja eesmärk ... sisuliselt siis nõuded mudelile, mida hakkate modelleerima)

2. Metoodika (umbes 20% tööst, kus kirjas millist notatsiooni ja miks kasutate; kus on kirjas ka muud vajalikud asjad, et kolmas osa oleks mõistetav)

3. Mudel ise (umbes 35% tööst, CASE vahendis, kus arusaadavas notatsioonis kirjas mudel)

4. Mudeli analüüs ja arutelu (võrdlus teiste analoogiliste mudelitega; kuidas see mudel muudatusi talub; kuidas see mudel UI-ga seostub; kuidas see mudel andmebaasiga seostub )

5. Kokkuvõte (1 lause probleemist, 1 lause eesmärgist, 2 lauset metoodikast, 2-3 lauset mudelist, 2-3 lauset mudeli analüüsist ja arutelust )