

## Eksamiküsimuste vastused aines Tarkvaratehnika (MTAT.03.094)

Aeg: 19. jaanuar 10:00 - 14:00

Küsimuste & vastuste redaktor: Oleg Mürk ([oleg.myrk@gmail.com](mailto:oleg.myrk@gmail.com))

Põhimõtted:

- Lühiküsimuste eesmärk on teadmiste kontroll. Vastused küsimustele leiab loengumaterjalidest.
- Lühiküsimuste hindamisel on ainus kriteerium vastaja demonstreeritud teadmised.
- Seega nt küsimusele "mis on muster Observer?" antud vastus "see on nagu Swingi listenerid" on täiesti piisav.
- Pikkade küsimuste eesmärk on hinnata vastaja oskused analüüsis, projekteerimises, ja testimises, mida sai omandada lahendades praktikumi ülesandeid.

1) (3 p.)

*Küsimus:*

Selgita väleda ning plaan-juhitud arenduse olemust. Millised on nende plussid ning miinused? Millal tuleks millist kasutada? Too näited väledast ning plaan-juhitud arenduse metoodikast.

*Materjal:*

[http://ats.cs.ut.ee/courses/2005/tvt/pmwiki/uploads/Main/software\\_engineering\\_07.pdf](http://ats.cs.ut.ee/courses/2005/tvt/pmwiki/uploads/Main/software_engineering_07.pdf)

*Vastus:*

Väleda arenduse puhul keskendutakse inimestevahelisele suhtlusele ning hetkeseisule - ei tehta põhjalikke plaane, pigem adopteerutakse olevikuga. Plaan-juhitud arenduse puhul on suurem rõhk dokumentatsioonile ning tegevuse planeerimisele, ehk tuleviku ennustamisele. Väledad metoodikad sobivad paremeni kui muudatuste hind on väike, nõuded muutuvad tihti, produktiivsus on tähtsam kvaliteedist. Plaan-juhitud metoodikad sobivad paremeni olukorda, kus muudatuse hind on kallis, nõuded muutuvad vähem, kvaliteet on tähtsam produktiivsusest. XP on väle, Waterfall plaan-juhitud, RUPi saab konfigureerida nii väledaks kui ka plaan-juhituks.

2) (3 p.)

*Küsimus:*

Kirjelda lühidalt kasutusloo analüüsi olemust ning kirjelda mis tingimustel selle kasutamine on põhjendatud. Koosta lühike näide.

*Materjal:*

[http://ats.cs.ut.ee/courses/2005/tvt/pmwiki/uploads/Main/software\\_engineering\\_08.pdf](http://ats.cs.ut.ee/courses/2005/tvt/pmwiki/uploads/Main/software_engineering_08.pdf)

Slaid 7

<http://ats.cs.ut.ee/courses/2005/tvt/pmwiki/uploads/Main/uml.zuml>

*Vastus:*

Kasutusloo analüüs kirjeldab rakenduse funktsionaalseid nõudeid. Kasutuslugu on rakenduse ja osaleja vaheline koostöö, mis omab täpselt määratud algust ning lõpeb selgelt mõistetava tulemusega osaleja jaoks. Kasutusloo analüüs koosneb kasutusloo skeemidest ja stsenaariumitest. Stsenaarium on süsteemi ja osaleja vaheliste interaktsioonide järjend, mis samuti defineerib erindsituatsioonid ning võimalikud lõpptulemused. Kasutusloo skeem on joonis, millel mullidega esitatakse kasutuslood, kriipskujudega – osalejad ja nooltega – seosed kasutuslugude ja osalejate vahel.

3) (3 p.)

*Küsimus:*

Kirjelda vähemalt 3 hea OO disaini põhimõtet. Põhjenda nende kasulikkust.

*Materjal:*

[http://ats.cs.ut.ee/courses/2005/tvt/pmwiki/uploads/Main/software\\_engineering\\_15.pdf](http://ats.cs.ut.ee/courses/2005/tvt/pmwiki/uploads/Main/software_engineering_15.pdf)

Slaidid 9-13

4) (3 p.)

*Küsimus:*

Kirjelda kolm peamist tarkvara arhitektuuri stiili. Too näited.

*Materjal:*

[http://ats.cs.ut.ee/courses/2005/tvt/pmwiki/uploads/Main/software\\_engineering\\_14.pdf](http://ats.cs.ut.ee/courses/2005/tvt/pmwiki/uploads/Main/software_engineering_14.pdf)

Slaidid 10

5) (3 p.)

*Küsimus:*

Defineeri tarkvarameetrika. Milleks tarkvarameetrikaid kasutatakse. Kirjelda 3 meetrika eritiüpi iseloomu järgi. Too näited. Nimeta 3 meetrika mõõtmisviisi. Too näited.

*Materjal:*

[http://ats.cs.ut.ee/courses/2005/tvt/pmwiki/uploads/Main/software\\_engineering\\_21\\_intro.pdf](http://ats.cs.ut.ee/courses/2005/tvt/pmwiki/uploads/Main/software_engineering_21_intro.pdf)

Slaidid 3, 9

6) (5 p.)

*Küsimus:*

Millist probleemi lahendab Object-Relational Mapping raamistik? Kirjelda ORM raamistiku toimimise põhimõtet. Nimeta vähemalt üks ORM raamistiku realiseerimine.

*Materjal:*

[http://ats.cs.ut.ee/courses/2005/tvt/pmwiki/uploads/Main/software\\_engineering\\_17.pdf](http://ats.cs.ut.ee/courses/2005/tvt/pmwiki/uploads/Main/software_engineering_17.pdf)

Slaidid 9-14

*Vastus:*

Object/Relational Mapping raamistik lahendab probleemi, mis seisneb selles, et programmeerija peab kirjutama rakenduse, mis funktsioneerib kahes maailmas: äri loogika on kirjutatud objekt-orienteeritult, aga andmete hoidmine ning pärimine relatsiooniliselt. O-o ja relatsiooniline esitus on erineva võimsusega - nt pärilus ei ole loomulikult väljendatav relatsioonilises vormis, samas relatsioonilises vormis on andmetega lihtsam ja kiirem manipuleerida. Tulemuseks on see, et andmetega töötamine toimub kasutades liidest, mis ei toeta objekt-orienteeritud programmeerimist (nt JDBC). Lisaks, kannatab produktiivsus, sest programmeerija peab tihti ümber lülitama kahe programmeerimisviisi vahel.

ORM raamistik pakub võimaluse töötada relatsiooniliste andmetega kasutades objekt-orienteeritud liidest, ehk kui programmeerija opereerib andmetega o-o kujul, siis samal ajal ORM raamistik teostab vastavad toimingud ka relatsioonilises vormis olevatel andmetel. Üldjuhul ORM raamistik paneb vastavusse klassid ja tabelid, objektid ja read ning isendi tunnused ja tabeli tunnused. ORM raamistik automaatselt tagab andmete vastavust erinevatel kujudel; võimaldab intellegentset andmete pärimist ja cache-mist; oskab erinevatel viisidel väljendada pärilust.

7) (10 p.)

*Autor:*

Ivo Mägi ([ivo@webmedia.ee](mailto:ivo@webmedia.ee))

*Ülesanne:*

Ainemudeli koostamine.

*Lahendus:*

Lahenduse üldnäite saab Arlow, Neustadt „Enterprise Patterns and MDA” raamatu 11 peatükist.

*Hindamine:*

Kõigi alltoodud punktide korral lähtuda sisulisest korrektsusest, mitte sellest et tudeng paberile formaalset korrektset UMLi ei ole suutnud joonistada. Oluline on kasutatava notatsiooni arusaadavus.

- Käsitletud peavad olema kõik loetletud olemid. (1pt)
- Igal olemil peavad olema välja toodud peamised tunnused (1pt)
- Olemitel mil eksisteerib käitumine peab see olema meetoditega välja toodud.  
Jälgida et oleks vähemalt üritatud mõelda low coupling ja high coherence printsiibi

järgi. (1+2pt)

- Olemitevahelised seosed peavad olema korrektsed (makse meetodi ja isiku puhul laiendamine, teiste puhul peamiselt seoste kordsus). Aggregationi ja/või compositioni demonstreerimise eest võib anda lisapunkti. (3pt)
- Selgelt peavad olema eristatavad liidesed ja/või abstraktsioonid. (2pt)
- Pakette võib aga ei pea kasutama (võib anda lisapunkti kui pakettide kasutamine on üldpilti lihtsustanud).

8) (10 p.)

*Autorid:*

Erik Jõgi ([erik.jogi@hansa.ee](mailto:erik.jogi@hansa.ee)),

Oleg Mürk ([oleg.myrk@gmail.com](mailto:oleg.myrk@gmail.com))

*Ülesanne:*

Koosta lihtsa graafikaredaktori disain.

*Lahendus:*

Model View Controller mustri kasutamine

Model - pilt ise

- pikselite raster
- pildi suurus

operatsioonid:

- etteantud tööriista ja parameetritega joonistamine (Command)
- pildi salvestamine/laadimine etteantud formaati tundjaga (Strategy)
- pildi suuruse küsimine

kirjeldus, kuidas on implementeeritud:

- interaktsioon tööriistadega – kuidas käsitletakse etteantud Command objekti (näiteks antakse neile ette java.awt.Graphics objekt, mille abil iga tööriist saab vajaliku kujutise pildile manada)
- interaktsioon salvestajatega – kuidas käitletakse etteantud Strategy objekti (erinevad strateegiad, kuidas pildi pikselitest koostada baidijada)

State – valitud joonistamise parameetrid

- valitud (ääre)joone värv
- valitud kujundite värv

Tööriistad on kõik sama interface'ga Command objektid:

- ikoon
- kursor?
- nimekiri parameetritest, mis tuleb määrata, et saaks vastava kujundi joonistada (näiteks ringil keskpunkt ja raadius, ristkülikul vasak ülemine ja parem alumine nurk)

- joonistamise meetod

Võiks olla välja toodud erinevate tööriistade implementatsioonide olulisemad punktid.

#### Controller

- valitud ala
- valitud tööriist

Observer muster: erinevad hiire tegevuste Listener'id. Oluline oleks nimekiri erinevatest listener'idest, mida tarvis läheb.

#### View

- Pildi kuvamine
- Tööriistade kuvamine
- Värvide valimine (kuidas toimub joone, kuidas kujundi värvi valimine?)
- Erinevad kursorid sõltuvalt hiire asukohast ja valitud tööriistast

Mis peaks veel olema välja toodud:

- Kuidas kontrolleri juhib tegevust?
- Erinevate formaatide salvestajate/laadijate interface
- Kirjeldus, kuidas tööriistadele vajalike punktide määramine käib
- Mis toimub rakenduse käivitamisel

#### *Hindamine:*

- Vastus peaks sisaldama
  - klassiskeem või muu viis rakenduse struktuuri kirjeldamiseks,
  - käitumist kirjeldavad skeemid või struktureeritud tekst,
  - selgitav tekst,
  - olulisemate meetodite pseudokood.
- Hinne peaks moodustuma proportsionaalselt ülaltoodud faktorite kaetusele ning esituse selgusele.
- Lisaks tuleb arvestada hea disaini põhimõtetega mis olid kirjeldatud loengus ning ülesande püstituses.
- Swing/AWT API ning UML süntaksi detailide tundmine ei tohiks mõjutada hinnet.

9) (10 p.)

*Autor:*

Härmel Nestra ([harmel.nestra@ut.ee](mailto:harmel.nestra@ut.ee))

*Ülesanne:*

Projekteerida testide komplekt shelli käsura testimiseks musta kasti meetodil.

*Lahendus:*

Järgnevalt on kirjeldatud testide jaotused ekvivalentsiklassideks paljude tunnuste

alusel. Konkreetne klass võib omakorda jaotuda klassideks. Sama tunnuse järgi võib klassideks mitut moodi jaotada.

Testide komplekt tuleks soovitatavalt teha selline, et kui iga klassijaotuse järgi valida suvaliselt üks klass, nii et valitud klasside ühisosa on mittetühi, siis leidub kompleksis test, mis kuulub sinna ühisossa. Niisugune testide komplekt tuleks liiga suur, et teda siia kirja panna, seepärast rahuldume klassijaotuste kirjeldamisega.

Kirjeldamisel on kasutatud minimalistlikku stiili. Tunnussõna järel on koolon ning selle järel on loetletud võimalik klassijaotus selle tunnuse alusel. Näiteks tunnus "Töö" jaotab võimalike testide ruumi klassideks kogu tehtava töö (st ühe testi ehk vajutatavate klahvide järjendi) omaduste järgi.

Töö:

- erifunktsioonklahvideta,
- erifunktsioonklahvidega.

Seis töö lõpul:

- kursor lõpus,
- kursor mujal.

Käsu ja argumentide seis töö lõpul:

- tühi,
- ainult käsk,
- käsk koos paljude argumentidega.

Käsk argumentidega:

- vahel üksikud tühikud,
- vahel juhuslik arv tühikuid.

Erifunktsioonklahvidega töö:

- erifunktsioonklahvidest ainult nooleklahve kasutav,
- erifunktsioonklahvidest ainult nooleklahve ja backspace'i kasutav,
- erifunktsioonklahvidest ainult nooleklahve ja delete'i kasutav,
- erifunktsioonklahvidest ainult tab'i kasutav,
- erifunktsioonklahvidest ainult nooleklahve ja tab'i kasutav,
- kõiki erinevaid erifunktsioonklahve kasutav.

Kasutab nooleklahve:

- ainult üht,
- mõlemat korra,
- korduvalt,

- rea alguses,
- rea lõpus,
- mujal.

Kasutab nooleklahve ja backspace'i:

- backspace rea alguses,
- backspace rea lõpus,
- backspace mujal,
- backspace peale nooleklahvivajutust.

Kasutab nooleklahve ja delete'i:

- delete rea lõpus,
- delete rea alguses,
- delete mujal.

Kasutab tab'i:

- esimeses sõnas,
- teises sõnas,
- väga kauges sõnas,
- rea lõpus.

Tab rea lõpus:

- rida on pikk,
- rida on tühi.

Kasutab nooleklahve ja tab'i:

- tab sõna algul,
- tab sõna keskel,
- tab viimases sõnas,
- tab sõnas, millele järgneb palju sõnu.

Kasutab kõiki erifunktsioonklahve:

- ei kasuta erinevaid erifunktsioonklahve järjest,
- kasutab paljusid eri pikkusega erifunktsioonklahvide järjendeid.

Vajutab tab'i:

- maksimaalselt üks kord järjest,
- maksimaalselt kaks korda järjest,
- palju kordi järjest,
- kui täiendusvõimalusi on 0,
- kui täiendusvõimalusi on 1,
- kui täiendusvõimalusi on 2,

- kui täiendusvõimalusi on väga palju.

Sõna:

- koosneb tähtedest,
- sisaldab mittetähti,
- ei vaja interpreteerimist,
- vajab interpreteerimist (jutumärgid, langjooned jm).

See jaotuste nimekiri ei pretendeeri täielikkusele. Praktika on näidanud, et tudengid leiavad ka selliseid mõistlikke jaotusi, mille peale mina pole tulnud.

Sellest pole midagi, kui tudeng pole nii lühidalt või selgelt osanud vastata kui on siinantud näidisvastus. Ka mina mõtlesin selle süsteemi alles näidisvastust koostades välja, see pole mingi klassikaline viis, mille ma oleks loengus vargsi enda teada jätnud.

*Hindamine:*

- Hinne peaks moodustuma proportsionaalselt ekvivalentsiklasside kaetusele.
- Konkreetsete testide koostamine ei ole tähtis, piisab klasside kirjeldamisest.