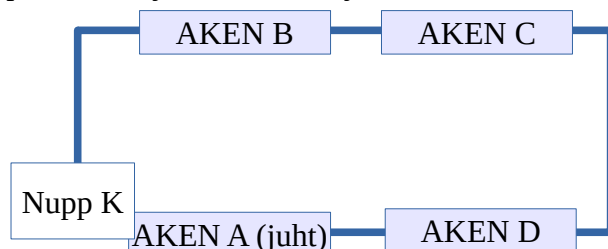


6. nädala praktikum IDK0051, kontrolltöö ettevalmistus

2015. a sügis

Töötate autotootja juures ning valmistute turule tooma uut autot. Sellel autol on 4 elektriakent (A, B, C ja D). Igat akent saab elektriliselt avada ja sulgeda. Lisaks on olemas kontrollnupp K, mis **lülitab osaliselt välja** akende C ja D mehhanismid – kui K on alla vajutatud, ei saa aknaid C ja D avada. Kontrollnupu K allavajutamine ei mõjuta eesmiseid aknaid A ja B.



Teie ülesanne on realiseerida tarkvara akende kontrollimiseks järgmise spetsifikatsiooni alusel.

Palun lugege enne alustamist kogu ülesanne läbi. Ülesannet ei pea lahendama punktides näidatud järjekorras!

1. Süsteemi testimiseks looge testklass, mis asuks **paketist väljaspool või eraldi paketis**.
2. Akende juhtimine toimub läbi nuppude, iga akna juures on nupp, mida saab lülitada erinevatesse asenditesse. Tarkvaraliselt te ei pea nuppe realiseerima, kuid võib-olla aitab see teadmine luua paremat ettekujutust süsteemist.
3. Aknal on järgmised meetodid, mida nupuvajutustega esile saab kutsuda:
 - täielik avamine
 - täielik sulgemine
 - osaline avamine 20% ulatuses
 - osaline sulgemine 20% ulatuses

Salvestage objektis akna olek täisarvuna – algolek (suletud) on 0 ning täielikult avatud olek on 10, see muutub vastavalt ülalnimetatud meetodite väljakutsumisele.

Lisaks kaks informatiivset meetodit:

- oleku tagastamine (tagastab täisarvu vahemikus 0 – 10, kus 0 on täielikult suletud ja 10 on täielikult avatud). Näide: algseis on alati 0 ning kui kaks korda avada akent 20% ulatuses, siis see meetod tagastab 4).
- meetod akna info tekstiliseks esituseks, sh millise aknaga tegu on (A ... D) ja oleku. See meetod peab **üle kirjutama ühe Object klassi meetodi**.

Ärireegel: akna olek ei tohi kunagi muutuda väiksemaks kui 0 ega suuremaks kui 10!

Süsteemi nõue: akna klass(id) peavad asuma eraldi paketis kui testklass.

Süsteemi nõue: Akna konstruktor(id) on protected või package-private.

Süsteemi nõue: arendage akna klass(id) TDD meetodil

4. Aknaid on kahte tüüpi:

- Eesmised (aknad A ja B)

- Tagumised (aknad C ja D)

Erinevus on selles, et kontrollnupuga K on võimalik tagumiste akende avamine blokeerida. Kui K on alla vajutatud, ei tohi tagumised aknad avaneda ega olekut muuta. Esimestele akendele nupp K mõju ei avalda.

Ärinõue: Iga avamise ja sulgemise operatsiooni järel trükitakse konsooli mida tehti ja akna olek.

Ärinõue: kui tegemist on aknaga C või D ning nupp K on sisse lülitatud, siis akna avamiskäsule ei reageerita ja trükitakse välja info „Akna avamine on hetkel blokeeritud” ning akna info, sh olek.

5. Looge kontroller, kus on *static factory method* aknaobjekti saamiseks. Argumendiks on akna tähis (A, B, C või D). Vajadusel või soovi korral looge mitu *static factory methodit*.

6. Kontrollerisse lisage meetod(id) nupu K oleku muutmiseks ja tagastamiseks. Nupp K saab olla sees või väljas. Kui ta on väljas, toimivad kõik aknad – kui sees, siis C ja D saab ainult sulgeda.

7. Testklassis looge kõigist akendest objektid (*static factory method* abil!), lisage need kollektsiooni ja itereerige seda kollektsiooni nii, et iga akna kohta:

- nupu K asend muudetakse OFF (väljas – kõik aknad töötavad piiranguteta)
- aken avatakse 20%
- aken avatakse täielikult
- aken avatakse 20%
- aken sulgetakse 20%
- aken sulgetakse täielikult
- aken sulgetakse 20%
- aken avatakse 20%
- nupu K asend muudetakse ON (sees)
- aken avatakse 20%
- aken sulgetakse täielikult

8. **Looge kahe elektriaknaga auto.** Lisage kontrollerisse *static factory method(id)*, mis tagastab akna asemel Optionali aknast (ärge muutke olemasolevaid meetodeid). Akende C ja D puhul peab tagastatav Optional tühi olema. Looge uus testklass kahe elektriaknaga automudeli jaoks ja looge aknad uute meetodite abil. Testklass peab töötama identselt olemasolevale testklassile – kõik neli Optionali objekti lisatakse kollektsiooni ja osalevad itereerimisel, kuid programm ignoreerib tsükli sees puuduvaid elektriaknaid.