



## POKERIKÄDET (vrt. kirjan tehtävät 36-42, s. 196)

Ensin ohjeita työympäristössä toimimiseksi. Tehtävät alkavat sivulta 3.

### Avaa etäyhteys Metropolian sisäverkkoon VPN-clientin avulla

Koneen voi liittää etäyhteydellä Metropolian sisäverkkoon VPN-yhteydellä (virtual private network). Tällöin sisäverkon palveluja voi käyttää Metropolian kampusten ulkopuoleltakin. Tällä kurssilla VPN-yhteyttä **tarvitaan aina SVN-versiohallinnan käytössä**.

Asennus: VPN-yhteyden saat käyttämällä Metropolian tarjoamaa Cisco AnyConnect VPN-clientiä. VPN-clientin saat asennettua kirjautumalla selaimella osoitteeseen <https://vpn.metropolia.fi>, jonka jälkeen valitse vasemmalta AnyConnect ja Start AnyConnect. Mikäli VPN-clientin automaattinen asennus ei onnistu, asenna client manuaalisesti lataamalla client napsauttamalla AnyConnect VPN -linkkiä. Lisätietoja etäkäytöstä Metropolian tietohallinnon sivuilta.


Käyttö: Kirjaudu VPN-clientilla (Windows: kuvake tehtäväpalkissa) osoitteeseen [vpn.metropolia.fi](https://vpn.metropolia.fi). Autentikointi ei käytä sertifikaattia, joten mahdollisen sertifikaattikyselyn voi ohittaa. Kun AnyConnect-yhteys on päällä, näkyvät sisäverkon palvelimet koneellesi.

### Tee Metropolian versionhallintapalvelimelle oma repositorio

- 1) kirjaudu Metropolian tunnuksellasi sivulle <https://innoscm-new.metropolia.fi/scm/>
- 2) napsauta Add Repository -painiketta
- 3) anna Name-kenttään repositoriosi nimi (esim. olso), se tulee repositorion URL-osoitteeseen.
- 4) valitse Type-valikosta Subversion ja ruksaa Initialize repository -valintaruutu.
- 5) syötä Contact-kenttään Metropolia sähköpostiosoitteesi
- 6) syötä Description-kenttään repositorion kuvaus, esim. Olso-kurssin harjoitustehtävät
- 7) napsauta Save-painiketta

Tekemäsi repositorion URL-osoite näkyy avautuneella Information-sivulla.


### Hae tehtäväsarjan koodit kurssin repositoriosta omalle koneellesi (export)

- 1) valitse Eclipsen oikeassa ylälaudassa olevasta Open Perspective -kuvakkeesta SVN Repository Exploring
- 2) napsauta SVN Repositories -välilehdeltä New Repository Location -kuvaketta 
- 3) syötä Url: <https://innoscm.edusrv.metropolia.fi/svn/olso>
  - syötä Metropolia-tunnuksesi ja salasanasasi
- 4) etsi haluamasi hakemisto **trunk/OlsoTehtSarja1** ja napsauta sitä hiiren oikealla
- 5) valitse **Export**
  - anna kohdehakemistoksi kurssilla käyttämäsi hakemisto, esim. C:\OlsoProjektit

### Avaa projekti Eclipsessä

- 1) valitse File | Open Projects from File System
- 2) napsauta Directory...-painiketta
- 3) valitse hakemisto OlsoTehtSarja1

## Vie projekti omaan repositorioosi (share project)

- 1) valitse SVN Repositories -välilehdeltä kuvake New Repository Location 
- 2) syötä Url, esim. <https://innoscm-new.metropolia.fi/scm/repo/omaTunnuksesi/olso/>
  - syötä Metropolia-tunnus ja salasanasasi niitä kysyttäessä
- 3) napsauta hiiren oikealla Package Explorerissa projektinimeä (ko. näkymään pääset Eclipsen oikean ylälaidan kuvakkeesta)
- 4) valitse Team | Share Project...
- 5) valitse SVN, ja tuplanapsauta oman reposi URL:ia
- 6) valitse täppä "Advanced Mode", tällöin trunk-hakemisto tulee polkuun (vakiintuneen käytännön mukaan trunk-hakemisto on työversioita varten).

Jos/kun tulee ilmoitus *"Unable to make field private java.lang.Throwable java.lang.Throwable.cause accessible: module java.base does not "opens java.lang" to unnamed module"*, niin sulje Eclipse ja lisää eclipse-hakemistossa olevan eclipse.ini -tiedoston loppuun rivit

```
--illegal-access=warn  
--add-opens java.base/java.lang=ALL-UNNAMED
```

Ja käynnistä Eclipse uudelleen.

Package Explorerissa näkyy projektin nimen perässä nyt oman reposi URL. Siitä tiedät, että työti-dostosi on kytkettynä versionhallintaan.

## Koodaa, testaa ja vie muutokset versionhallintaan (commit)

- 1) valitse Eclipsen oikeasta yläkulmasta **Java-perspektiivi (eli Package Explorer)**
- 2) koodaa ja testaa.

Muuttamiesi tiedostojen edessä näkyy väkäsmerkki (>). Kun olet tyytyväinen muutoksiisi, vie muutetut tiedostot versionhallintaan:

- 1) napsauta hiiren oikealla Package Explorerissa kommitoitavan tiedoston tunnusta
- 2) valitse Team | Commit...

Voit kommitoida myös koko pakkauksen tai koko projektin vastaavalla tavalla, napsauta tällöin pakkauksen tai projektin tunnusta.

Jotta palvelimella olevaan repositorioon tulleet uudet muutokset näkyisivät myös jo avoinna olevassa web-selaajassa, tyhjennä selaajan välimuistipuskurit. Esimerkiksi Firefoxissa se onnistuu F5-painik-keella.

**TOIMI TULEVISSAKIN HARJOITUKSISSA TÄTÄ OHJETTA SOVELTAEN.**

## TEHTÄVÄSARJA 1: POKERIKÄDET (vrt. kirjan tehtävät 36-42, s. 196)



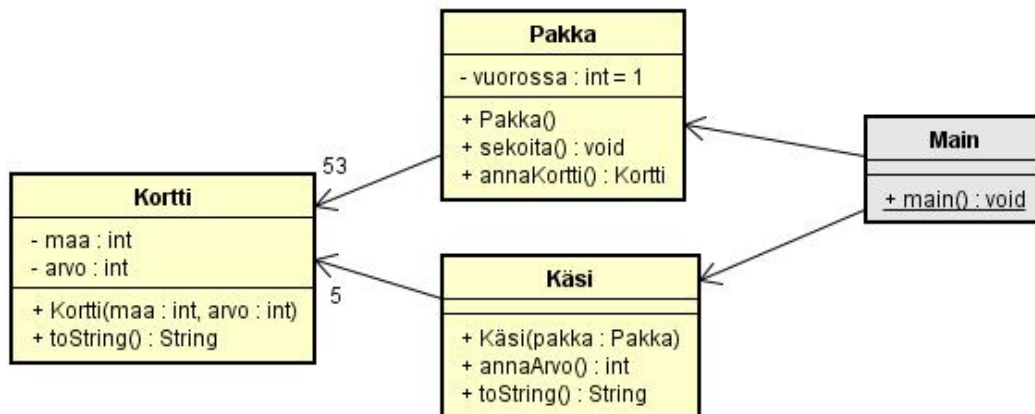
### Tavoite

Palauttaa mieleen taulukoiden käyttöä, ja syventää oliopohjaisen ohjelman laadintaitoa.

### Käsitteitä

luokka, olio, kapselointi, instanssimuuttuja, staattinen muuttuja (eli luokkamuuttuja), konstruktori, ylikirjoittaminen, kuormittaminen, getterit ja setterit, taulukko

Kurssin versionhallinnasta poimimassasi projektissa (OlsoTehtSarja1) on tehtävän luokkien tyngät, mutta koodi puuttuu. Tests-hakemistossa olevat JUnit-testit eivät edes käänny ennen kuin olet kirjoittanut puuttuvat koodit.



1. Ohjelmoi luokka **Korti**. Luokalla on kortin ominaisuudet instanssimuuttujissa `maa` (0=HERTTA, 1=RUUTU, 2=RISTI, 3=PATA) ja `arvo` (välillä 1..13). Kirjoita kahden parametrin konstruktori (järjestys: `maa`, `arvo`) ja julkiset saantimetodit (getterit: `getMaa()` ja `getArvo()`) sekä ylikirjoita `toString()`-metodi siten, että se palauttaa merkkijonona kortin maan ja arvon välilyönnillä eroteltuna (esimerkki: "Hertta 12").

Aja testit (KorttiTest.java): etsi Package Explorer-välilehdeltä tiedosto KorttiTest.java. Napsauta sitä hiiren oikealla ja valitse Run As | JUnit Test. Jos olet ohjelmoinut Kortti-luokan oikein, JUnit-välilehdellä pitäisi näkyä sen merkinä vihreä palkki. Jos palkissa on punaista, korjaa koodiasi ja aja testit uudestaan.

Kommitoi. Kun olet tyytyväinen tulokseen, vie muutokset versionhallintaan omaan repoosi:

- napsauta hiiren oikealla tiedostonimeä Kortti.java
- valitse Team | Commit...
- muista kirjoittaa hyvin harkittu ja muutoksia kuvaava kommentti

2. Ohjelmoi luokka **Pakka**. Luokan instanssimuuttujana on 53-alkioinen taulukko (ei ArrayList), jossa kukin alkio viittaa Kortti-olioon (luo 53 alkion taulukko, jätä lokero 0 käyttämättä). Kirjoita konstruktori, joka alustaa taulukon luomalla sinne Kortti-oliot. Täytä pakka järjestyksessä: luo ensin kaikki hertat suuruusjärjestyksessä, sitten ruudut, ristit ja padat.
3. Lisää luokkaan **Pakka** metodi `void sekoita()`, joka valitsee pakasta 100 kertaa satunnaisesti kaksi korttia ja vaihtaa niiden paikat keskenään.

Tee myös metodi `Korti annaKortti()`, joka palauttaa pakasta seuraavana jakovuorossa olevan kortin. Pidä kirjaa jakovuorossa olevasta kortista instanssimuuttujassa `vuorossa`. Ensimmäisellä kerralla jakovuorossa on indeksillä 1 oleva kortti, seuraavalla kutsukerralla 2, jne. Kun on annettu pakan viimeinen kortti, aloitetaan taas alusta eli kortista 1.

Aja testit (PakkaTest.java) tehtävässä 1 annetuin ohjein. Tarkasta itse JUnitin tulosteista, että sekoitus toimii oikein (sekoitusta koskeva testi on laitettu tuottamaan aina punaista).

Kommitoi.

4. Ohjelmoi luokka **Käsi**. Luokka sisältää instanssimuuttujana taulukon, jonka alkiot viittaavat viiteen Kortti-olioon. Luo konstruktori, joka saa parametrina pakan. Konstruktori pyytää pakasta `annaKortti()` -metodilla viisi korttia, ja täyttää taulukon metodin palauttamilla Kortti-olioilla. Kirjoita myös `toString()` -metodi, joka palauttaa merkkijonoesityksen käden korteista (hyödynnä Kortti-luokan vastaavaa metodia), esimerkiksi "Hertta 1, Ruutu 3, Hertta 12, Risti 4, Pata 3"

Kirjoita luokkaan **Käsi** metodi `int annaArvo()`. Sen tulee palauttaa arvo 2 (eli vakio VÄRI), jos käsi sisältää värin (ts. kaikki kortit ovat samaa maata) ja muutoin arvo 0.

**Huom: Jatka seuraavaan tehtävään.** JUnit-testeissä käytetään tehtävän 6 konstruktoria.

5. Ohjelmoi oletuspakkaukseen (Eclipsen default) luokka **Main**, jossa on metodi `main()`. Se luo ja sekoittaa pakan, jakaa käden ja näyttää sen käyttäjälle sekä tarkastaa tuliko väri. Kirjoita ympärille toistorakenne, joka jatkaa käsien jakoa siihen saakka, kunnes väri saadaan. Sekoita pakka kullakin kierroksella. Tulosta lopuksi kuinka monta kättä täytyi jakaa värin saavuttamiseksi (vihje: lisää staattinen muuttuja ja sille getteri Käsi-luokkaan).

Tehtävät 1-5 muodostavat oman kokonaisuutensa, ja ainakin ne tulisi ratkoa. Myös seuraavat tehtävät kuuluvat tehtäväsarjaan, ja niistä kertyy pisteet kuten edellä olevistakin.

### Jatkotehtäviä

6. Halutun korttikombinaation aikaan saaminen testaamista varten on vaikeaa (random). Lisää testaamisen helpottamiseksi luokkaan Käsi toinen konstruktori, jolle voi antaa parametrina viisi Kortti-oliota sisältävän taulukon (josta käsi muodostetaan - ei siis pyydetä pakasta kuten tehtävässä 4). KäsiTest-luokka käyttää tätä konstruktoria.

Aja testit (KäsiTest.java) vastaavalla tavalla kuin aiemmat testit.

Kommitoi.

Voit ajaa kaikki tähän tehtäväkertaan liittyvät testit yhdellä käynnistyksellä: napsauta hiiren oikealla Package Explorerissä tests-hakemistoa ja valitse Run As | JUnit Test.

Kurkista testiluokkien koodeja. Tutustumme JUnit-testien laatimiseen myöhemmin kurssilla.

7. Paranna metodia `int annaArvo()` siten, että se palauttaa arvon 1 (eli vakio SUORA), jos käsi sisältää suoran ja arvon 3 (eli vakio VÄRISUORA), jos kädessä on värisuora. Suora tarkoittaa tilannetta, jossa käden korteilla on viisi peräkkäistä arvoa (esim. 4, 5, 6, 7, 8). Huomaa, että ässä on joko 1 tai 14 sen mukaan, kumpi on edullisempi. Värisuora on kyseessä silloin, jos kädessä on samanaikaisesti sekä väri että suora. Ps.  $2+1=3$ ,  $VÄRI+SUORA=VÄRISUORA$ .

Luokassa KäsiTest on valmiina JUnit -testit, joilla voit varmistua siitä, että metodi `annaArvo()` arvioi oikein myös suoran ja värisuoran. Poista tai kommentoi testimetodien edessä olevat `@Disabled`-annotaatiot. Aja testit. Kun testit menevät läpi, toteutuksesi on kunnossa.

Muokkaa vielä luokkaa Main.java siten, että käsien jako jatkuu, kunnes saadaan joko suora, väri tai värisuora. Mitä voit päätellä tarvittavien jakojen määrästä?

Kommitoi.