

## GUI - Java FX

### Tavoite

*Oppia kirjoittamaan MVC-mallin mukaisia yksinkertaisia JavaFX-sovelluksia.*

### Käsitteitä

*MVC-malli, tapahtumaohjattu sovellus, ikkuna, näkymä, asettelupohja, kontrollit*

**Alkutoimet:** Luo Eclipse-projekti kalvosarjan ohjeita soveltaen (JavaFX Project). Toteuta seuraaviin sovelluksiin graafinen käyttöliittymä ja testaa koodisi. Tehtäviin ei liity JUnit-testejä, mutta voit tehdä niitä halutessasi. Vie projektisi svn:ään nimellä **OlsoTehtSarja8**.

1. Kirjoita ohjelma, jonka käyttöliittymässä on selite, jossa lukee "Napsauta painiketta". Käyttöliittymässä on myös neljä painiketta ja niissä tekstit Aamu, Päivä, Ilta ja Yö. Kun napsautat painiketta, selitteen sisällöksi vaihtuu vastaava tervehdys, esimerkiksi Hyvää huomenta. Sijoita tehtävän koodi JavaFX:n generoimaan **application**-pakkaukseen (esimerkiksi nimelle **TervehdysGUI.java**). Elementtien sijoittelun voit valita haluamaksesi.



2. Meille tutuimmassa nopassa, arpakuutiossa, on 6 sivua (d6) ja kullakin sivulla on yksikäsitteinen silmäluku 1-6. Sivuja voi yhtä hyvin olla joku muukin määrä (tehtävän 5 kolikkotehtävässä 2 sivua, d2). Ks. <https://fi.wikipedia.org/wiki/Noppa>

Tee **model**-niminen pakkaus ja laadi sinne luokka Noppa.java, joka sisältää metodit

- Noppa(int silmäluku) konstruktori, jolla parametrina nopan sivujen lukumäärä
- heita() palauttaa nopasta satunnaisesti valitun silmäluvun

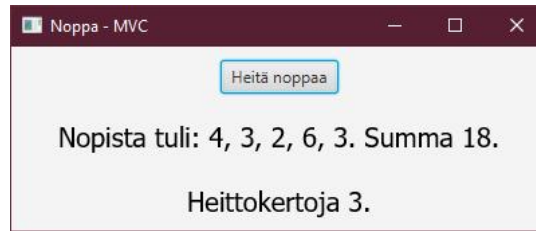
Tee **controller**-niminen pakkaus, jonne tulee koodi, joka yhdistää käyttöliittymän ja mallin (model).

Toteuta sitten **MVC-mallin** mukainen konenoppasovellus yhdelle kuusisivuiselle nopalle (eli arpa-kuutiolle). Sijoita käyttöliittymän koodi application-pakkaukseen (esim. nimelle **NoppaGUI.java**). Käyttöliittymässä on vain yksi painike ja yksi tekstikenttä. Elementtien sijoittelun voit valita haluamaksesi.

3. Laajenna sovellusta siten, että se pitää kirjaa heittojen lukumäärästä ja näyttää silmäluvun ohella siihen mennessä tehtyjen heittojen lukumäärän. Tässä halutaan, että lukumäärä on noppaan liittyvä tieto. Lisää Noppa-luokkaan myös metodi getHeittojenLkm().



4. Laajenna sovellusta niin, että yhden nopan sijasta yksi painikkeen napsautus heittää viittä eri noppaa, ja näyttää käyttäjälle silmälukujen summan.



Huomaa, että on eri asia heittää yhtä noppaa viisi kertaa, kuin heittää viittä noppaa "yhtä aikaa". Jos nopalta kysyy, montako kertaa sitä on heitetty, niin kaikki viisi noppaa palauttaisivat saman lukumäärän. Määrittele Controller-tasolle sopiva tietorakenne viiden nopan hallitsemiseksi. Älä tee käyttöliittymään (View) tai noppaan (Model) muutoksia. Yllä siihen ainoaan tekstikenttään on viety teksti "Nopi sta tuli : 4, 3, 2, 6, 3. Summa 18. \n\nHei ttokertoja 3."

5. Kuten tehtävä 2, mutta nyt ohjelmassa heitetään kolikkoa, eli 2-sivuista noppaa (d2) ja tulostetaan heiton tulos numeron sijasta tekstinä "Kruuna" tai "Klaava". Sijoita käyttöliittymä application-pak-  
kaukseen (esim. nimelle **KolikkoGUI.java**). Mikä osa (M? V? C?) muuttuu verrattuna edelliseen tehtävään?

Kirjoita rajapinta **IView**, jossa esittelet metodit, joita kontrolleri-olio käyttää tulosten näyttämisessä. Kukin erilainen käyttöliittymä (tehtävä 2, tehtävä 5) sitten vain toteuttaa rajapinnan metodit omalla tavallaan. Korjaa tehtävien 2 ja 5 koodia käyttämään tätä rajapintaa.

6. Tee noppatehtävään toinen käyttöliittymä, jossa kuusisivuisen arpakuution silmäluku näytetään numeron sijasta vastaavaa sivua esittävänä kuvana. Lataa tarvittavat kuusi kuvatiedostoa Internetistä. Löydät tiedostot helposti hakukoneella hakusanoilla dice faces. Vihje: tutustu **ImageView**-kontrolliin.

