

## KAHVILA (vrt. kirjan tehtävät 43-49, s. 197)



### Tavoite

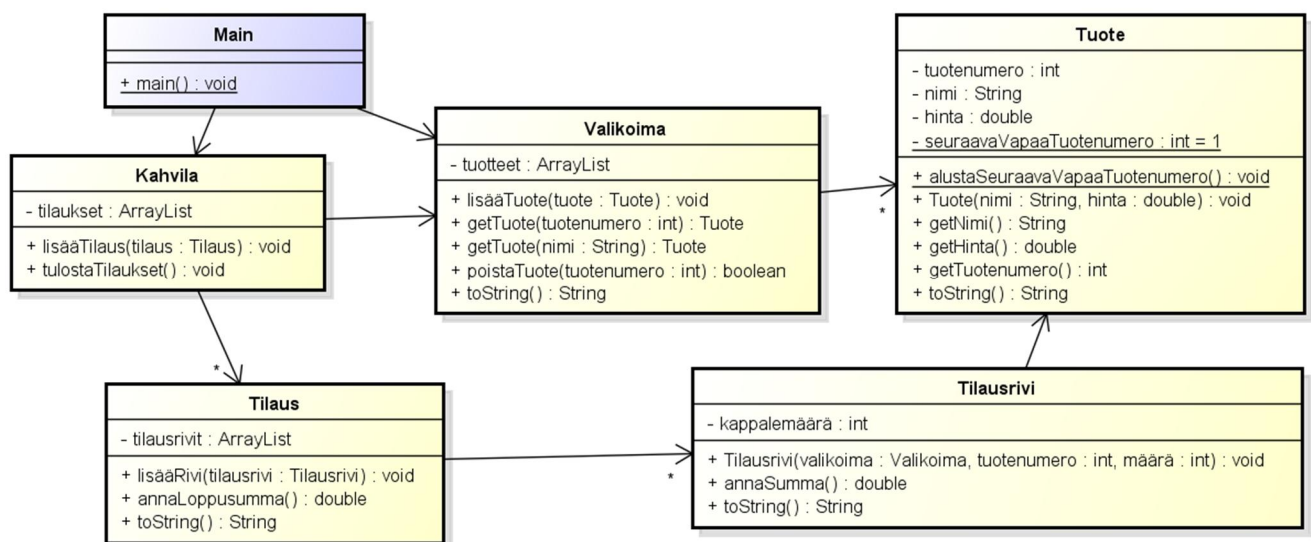
*Oppia laatimaan java.util-pakkaukseen kuuluvaa ArrayList-tietorakenneluokkaa hyödyntävä, useasta luokasta koostuva olioperustainen ohjelma.*

### Käsitteitä

*API (application program interface), import, ArrayList, for-each -rakenne*

**Alkutoimet:** Kopioi tehtäväsarjan koodit (**OlsoTehtSarja2**) kurssin tietovarastosta (repository) omaan repositorioosi ja luo Eclipse-projekti edellisen tehtäväsarjan ohjeita noudattaen.

Tehtävien kohdealueena on kahvila, jolla on myytävänä eri valikoimiin kuuluvia tuotteita. Valikoima voisi olla vaikkapa juomat, leivokset, karamellit, tms. Kukin kahvilaan tuleva asiakas tekee kassalla tilauksen, joka koostuu tilausriveistä. Kukin tilausrivi kertoo, minkä valikoiman tuotetta asiakas ostaa ja kuinka monta kappaletta.



1. Ohjelmoi luokka **Tuote**, jonka instanssimuuttujia ovat tuotenumero, nimi ja hinta. Toteuta metodit

- `public Tuote (String nimi, double hinta)`
- `public String getNimi()`
- `public double getHinta()`
- `public int getTuotenumero()`
- `public String toString()`
- `public static void alustaSeuraavaVapaaTuotenumero()`

Aseta tuotenumeron konstruktorissa. Määrittele staattinen luokkamuuttuja seuraavan vapaan tuotenumeron säilyttämiseksi. Ensimmäinen tuotenumero on 1. Metodilla `alustaSeuraavaVapaaTuotenumero()` voi asettaa seuraavan vapaan tuotenumeron alkuarvoonsa. Ohjelmoi `toString()` palauttamaan tiedot esimerkiksi muodossa: "1: Dallaspulla, 1.55€".

Aja testit (TuoteTest.java) edellisen tehtäväsarjan ohjeiden mukaisesti. Siirry seuraavaan tehtävään vasta, kun kaikki testit menevät läpi.

2. Ohjelmoi luokka **Valikoima**, joka sisältää instanssimuuttujana Tuote-olioista koostuvan ArrayList-olion, ts. valikoima pitää kirjaa yhdestä tai useammasta valikoimaan kuuluvasta tuotteesta. Toteuta metodit

- `public void lisääTuote(Tuote tuote)`
- `public Tuote getTuote(int tuotenumero)`
- `public Tuote getTuote(String nimi)`
- `public boolean poistaTuote(int tuotenumero)`
- `public String toString()`

Varmista myös, että tuotetta ei lisätä listaan useampaan kertaan. Jos tuotenumerolla tai -nimellä ei ole tuotetta, niin `getTuote()` palauttaa null. `poistaTuote()` palauttaa false, jos Valikoimassa ei ollut tuotetta annetulla tuotenumerolla. Muotoile `toString()` siten, että tuotteiden välissä on rivivaihto (eli merkki `\n`).

Aja JUnit-testi (ValikoimaTest.java).

- Ohjelmoi luokka **Tilausrivi**, jonka instanssimuuttujina on yksi Tuote-olio, johon tilausrivi liittyy sekä tilatun tuotteen kappalemäärä. Toteuta metodit:

- `public Tilausrivi(Valikoima valikoima, int tuotenumero, int määrä)`
- `public double annaSumma()`
- `public String toString()`

Hae konstruktorissa tuotenumeroa vastaava Tuote-olio valikoimasta kohdassa 2 kirjoittamallasi `getTuote()`-metodilla. Metodi `annaSumma()` palauttaa tämän tilausrivin kokonaishinnan (eli hinnan kerrottuna kappalemäärällä). Metodi `toString()` palauttaa merkkijonona tähän tilausriviin liittyvät tiedot.

Aja JUnit-testi (TilausriviTest.java).

- Ohjelmoi luokka **Tilaus**, joka sisältää instanssimuuttujana Tilausrivi-olioista koostuvan ArrayList-olion, ts. tilaus pitää kirjata yhdestä tai useammasta tilausrivistä. Toteuta metodit:

- `public void lisääRivi(Tilausrivi tilausrivi)`
- `public double annaLoppusumma()`
- `public String toString()`

Metodi `annaLoppusumma()` palauttaa tilauksen kokonaishinnan.

Aja JUnit-testi (TilausTest.java).

- Ohjelmoi luokka **Kahvila**, jonka instanssimuuttujana on valikoima sekä Tilaus-olioista koostuva ArrayList-olio, ts. kahvila pitää kirjata yhdestä tai useammasta tilauksesta. Toteuta metodit:

- `public Kahvila(Valikoima valikoima)`
- `public void lisääTilaus(Tilaus tilaus)`
- `public String toString()`

`toString()`-metodi palauttaa merkkijonon, josta käy ilmi mitä on tilattu sekä yhteishinta.

Aja JUnit-testi (KahvilaTest.java). Voit ajaa kaikki tähän tehtäväkertaan liittyvät testit napsauttamalla hiiren oikealla Package Explorerissa hakemistoa test ja valitsemalla Run As | JUnit test.

- Ohjelmoi oletuspakkaukseen (default package) luokka **Main**, jossa luot valikoiman sekä kahvilan, joka tarjoaa tuon valikoiman.
  - Luo ja lisää kolme tuotetta valikoimaan.
  - Tulosta valikoima ja poista siitä yksi tuote. Tulosta muuttunut valikoima.
  - Luo tilaus, luo ja lisää kaksi tilausriviä tilaukseen ja lisää tilaus kahvilan tilauksiin.
  - Luo toinenkin tilaus tilausriveineen.
  - Tulosta kaikki kahvilan tilaukset.

### Jatkotehtävä

- Toteuta edellisen kohdan `main()`-metodiin loppuun seuraavat toiminnot.

1. Lisää tuote valikoimaan
2. Tulosta valikoima
3. Poista tuote valikoimasta
4. Kirjaa tilaus
5. Tulosta kaikki tilaukset
0. Lopeta

Käyttäjä saa suorittaa toimintoja, kunnes hän haluaa lopettaa. Tarvitset siis noiden ympärille toistorakenteen. Ohjelma kysyy tarvittaessa tarkemmat tiedot, esimerkiksi tuotetta valikoimaan lisättäessä tuotteen nimen ja hinnan.