

Palauta kaikkien tehtävien ratkaisut Moodlen yhtenä tiedostona txt-muodossa.

Tehtävien ratkaisemisessa ei saa käyttää ChatGPT:tä tai vastaavia generatiivisen tekoälyn työkaluja.

Palauta ratkaisusi tiistaina 23.1 klo 23:50 mennessä.

Tehtäviä suositellaan tehtäväksi yhdessä toisten opiskelijoiden kanssa, mutta jokainen palauttaa omat itse tekemänsä ratkaisut.

Kunkin tehtävän lopussa on kerrottu siitä saatava pistemäärä. Pisteiden summa on 17 pistettä

Tavoitteena on *opiskella olio-ohjelmoinnin perusasioita*.

Ohjelmissa harjoitellaan

- yksinkertaisen olioluokan tekemistä: kenttiä, alustaja ja metodeja
- olioluokkaa käyttäjän pääohjelman tekemistä
- olioiden suojaamista määrittelemällä kenttiä yksityisiksi (kapselointi, private)

Asenna itsellesi IntelliJ Java-ohjelmointiympäristö (tai jGrasp tai komentoriviltä käytettävä Java) ja tee seuraavat tehtävät. Voit tehdä tehtävät myös laitoksen mikroluokassa. Saat neuvoja tehtävien tekemiseen mikroluokassa torstain ohjaustunneilla, läksykerhossa sekä kurssin Discord kanavilla sekä maanantain kyselytunnilla.

Tehtävissä on pyritty kertomaan varsin yksityiskohtaisesti mitä on tarkoitus tehdä. Lue tehtävät huolellisesti ja kirjoita ohjelmakoodi ohjeiden mukaisesti. Käytä tehtävissä annettuja (yleensä **tummennettuja**) luokkien, kenttien ja metodien nimiä. Käytetyt käsitteet selviävät kun tutustut viikon opiskelumateriaaleihin.

1. Suunnittele ja toteuta luokka **Biisi**, jossa on neljä kenttää: **nimi** (biisin nimi), **esittaja** (biisin esittäjä) ja **spotifySoittomaara** joka on kappaleen Spotifyn:n soittokertojen määrä. Luokassa on seuraavat metodit: (1 piste)
 - a. Kirjoita kaikille kentille **set-metodit**
 - b. Kirjoita kaikille kentille **get-metodit**
 - c. Kirjoita luokalle **alustaja**

Tee myös **testipääohjelma**, jossa luot luokasta kolme oliota luokasta, joiden tiedot voit keksiä. Tulosta lopuksi näiden olioiden kaikkien kenttien tiedot näytölle.

2. Tarkastellaan alla olevaa luokan määrittelyä. (2 pistettä)

```
public class Laskuri {  
    public int luku1, luku2;  
    public String s1;  
    public int kerro() {  
        int tulo = luku1 * luku2;  
        return tulo;  
    }  
}
```

Kopioi tämä koodi lähtökohdaksi ja sitten täydennä luokan määrittelyä seuraavasti:

- Tee luokalle **parametriton alustaja**, joka sijoittaa kenttien luku1 ja luku2 arvoksi 0.
- Kirjoita luokalle **alustaja**, joka saa parametrina kaksi kokonaislukua ja merkkijonon. Alustaja sijoittaa parametrit kenttien luku1, luku2 ja s1 arvoiksi.
- Kirjoita **pääohjelma**, joka
 - Luo luokasta **Laskuri** olion, jonka kenttien arvot ovat 23, 29 ja "Moikka".
 - Tulosta olion kenttien tiedot näytölle.
 - Laske metodin **kerro** avulla kenttien luku1 ja luku2 tulo ja tulosta sen näytölle.
 - Tulosta kentän s1 arvo näytölle.

3. Tarkastellaan alla olevaa ohjelmakoodia. (2 pistettä)

```
public class Juttu {  
    private int x;  
    private int y;  
    private static int z = 0;  
  
    public Juttu() {  
        x = z;  
        y = z;  
    }  
    static void asetaJuttu(int a) {  
        z = a;  
    }  
    public static void main(String [] args) {  
        Juttu yksi = new Juttu();  
        Juttu kaksi = new Juttu();  
        Juttu kolme = new Juttu();  
    }  
}
```

- Kuinka monta erillistä kenttää x ohjelmassa on luotu?
- Kuinka monta erillistä y kenttää ohjelmassa on luotu?
- Kuinka monta erillistä z kenttää ohjelmassa on luotu?
- Mitkä ovat x:n ja y:n arvot kussakin olion ilmentymässä?
- Kirjoita lause, jossa kutsut metodia asetaJuttu.

4. Suunnittele ja toteuta luokka **Suorakulmio**, jonka käsittelee geometrisen suorakulmion tietoja seuraavasti: (3 pistettä)
- Luokassa on kaksi double tyyppistä kenttää **leveys** ja **korkeus**, joissa käsitellään suorakulmion leveyttä ja korkeutta. Kummankin oletusarvo on 1.
 - Luokassa on parametrin alustaja, joka luo oletussuorakulmion.
 - Luokassa on alustaja, joka saa parametreina suorakulmion leveyden ja korkeuden.
 - Luokassa on metodi **getAla()** joka palauttaa suorakulmion pinta-alan
 - Luokassa on metodi **getYmpärys()**, joka palauttaa suorakulmion ympärysmittan.
 - Kirjoita luokalle testipääohjelma, joka luo kaksi Suorakulmio oliota. Toisen leveys on 4 ja korkeus 40. Toisen leveys on 3.5 ja korkeus 35.9. Tulosta näytölle kummankin suorakulmion leveys, korkeus, pinta-ala ja ympärysmitta tässä järjestyksessä.
5. Suunnittele ja toteuta luokka **Osake**, joka käsittelee pörssiosakkeiden tietoja seuraavasti: (3 pistettä)
- Luokassa on String tyyppinen kenttä, jonka nimi on **symboli**. Se sisältää osakkeen lyhenteen.
 - Luokassa on String tyyppinen kenttä, jonka nimi on **nimi**. Se sisältää osakkeen nimen.
 - Luokassa on double tyyppinen kenttä, jonka nimi on **edellinenHinta**. Se sisältää osakkeen edellisen päivän hinnan.
 - Luokassa on double tyyppinen kenttä, jonka nimi on **nykyinenHinta**. Se sisältää osakkeen nykyisen hinnan.
 - Luokassa on alustaja, joka luo olion saamansa symbolin ja nimen perusteella.
 - Luokassa on metodi **getMuutosProsentti()**, joka palauttaa hinnan muutoksen edellisen ja nykyisen hinnan välillä prosentteina.
 - Kirjoita testipääohjelma, joka luo Osake olion, jonka symboli on NOK ja nimi on Nokia OYJ, edellisen päivän hinta on 34.5 ja nykyinen hinta 34.35, ja se näyttää osakkeen tiedot sekä hinnan muutoksen prosentteina.

6. Suunnittele ja toteuta luokka **Tili**, joka käsittelee pankkitilin tietoja seuraavasti: (3 pistettä)
- a. Luokassa on yksityinen int tyyppinen kenttä **id**, joka sisältää tilinumeron (oletus 0).
 - b. Luokassa on yksityinen double tyyppinen kenttä **saldo**, joka sisältää tilin tämänhetkisen saldon (oletus 0).
 - c. Luokassa on yksityinen double tyyppinen kenttä **vuosiKorko**, joka sisältää tilin nykyisen koron (oletus 0). Oletetaan että kaikilla tileillä on sama korko.
 - d. Luokassa on yksityinen Date tyyppinen kenttä **luontiPaiva**, joka sisältää päivämäärän, jolloin tili on luotu.
 - e. Luokalla on parametrin alustaja, joka luo oletusolion.
 - f. Luokalla on alustaja, joka luo olion tietyllä **tilinumerolla** ja **saldolla**.
 - g. Luokalla on get- ja set-metodit, joilla voidaan katsoa ja muokata kenttien **id**, **saldo** ja **vuosiKorko** arvoja.
 - h. Luokalla on get-metodi, jolla voidaan katsoa kentän **luontiPaiva** arvo.
 - i. Luokalla on metodi **getKuukausiKorkoProsentti()**, joka palauttaa kuukauden korkoprosentin (millä prosentilla kuukauden aikana maksetaan korkoa, oletetaan että kaikissa 12 kuukaudessa on sama määrä päiviä).
 - j. Luokalla on metodi **getKuukausiKorko()**, joka palauttaa kuukauden aikana kertyneen koron määrän. Laske tämä kuukausikorkoprosentin avulla.
 - k. Luokalla on metodi **nosta**, joka nostaa annetun summan tililtä.
 - l. Luokalla on metodi **talleta**, joka tallettaa annetun summan tilille.
 - m. Kirjoita testipääohjelma, joka luo Tili olion, jonka ID on 1122 ja saldo 20000 ja vuosikorko 4.5 %. Nosta tililtä 2500 ja talleta 3000. Tulosta tilin saldo, kuukausikorko ja päivämäärä, jolloin tili on luotu.

7. Suunnittele ja toteuta luokka **Tuuletin**, joka käsittelee tuulettimen toimintaa seuraavasti: (3 pistettä)
- a. Luokassa on kolme vakiota: **HIDAS**, **NORMAALI**, **NOPEA**, joilla on arvot 1, 2 ja 3. Nämä kuvaavat tuulettimen nopeutta.
 - b. Luokassa on yksityinen int tyyppinen kenttä **nopeus**, joka määrittää tuulettimen tämänhetkisen nopeuden (oletusarvo HIDAS).
 - c. Luokassa on yksityinen boolean tyyppinen kenttä **kytketty**, joka määrittelee, onko tuuletin kytkettynä päälle vai ei. Oletusarvo false.
 - d. Luokassa on yksityinen double tyyppinen kenttä **halkaisija**, joka määrittelee tuulettimen halkaisijan, oletusarvo 5.
 - e. Luokassa on kenttä **vari**, joka määrittelee tuulettimen värin, oletusarvo blue.
 - f. Luokassa on get- ja set-metodit kaikille neljälle kentälle.
 - g. Luokassa on parametruton alusta.
 - h. Luokassa on metodi **toString()**, joka palauttaa tuulettimen kuvauksen. Jos tuuletin on kytkettynä, niin metodi palauttaa nopeuden, värin ja säteen, jotka on yhdistetty yhdeksi merkkijonoksi. Jos tuuletinta ei ole kytketty päälle, niin metodi palauttaa värin ja halkaisijan, sekä merkkijonon "tuuletin ei ole päällä" – yhdistettynä yhdeksi merkkijonoksi.
 - i. Kirjoita luokalle testipääohjelma, joka luo kaksi Tuuletin oliota. Aseta ensimmäiselle tuulettimelle maksiminopeus, halkaisijaksi 10, väriksi keltainen ja kytke tuuletin päälle. Aseta toiselle tuulettimelle normaali nopeus, halkaisijaksi 5, väriksi blue ja kytke tuuletin pois päältä. Tulosta olioiden tiedot näytölle toString-metodin avulla.