

Olio-ohjelmoinnin perusteet – Demo 2

Huom! Kaikkiin luokkiin tulee kirjoittaa tarvittavat konstruktorit ja asetus- ja havainnointimetodit attribuuteille.

1. Toteuta edellisten demojen Laiva-luokan pohjalta seuraavat uudet Laivan perivät luokat:
 - a. **Rahtilaiva:** uusina attribuutteina kapasiteetti (*double*) ja rahti (*double*). Rahtilaivaan ei luonnollisesti voi lisätä kapasiteettia enempää rahtia. Lisäksi rahdin määrä vaikuttaa laivan nopeuteen: jokainen täysi kymmenesosa kapasiteetista laskee laivan nopeutta 10% (eli täysi lasti laskee nopeutta 100% ja ½ lastia 50%), mikä pitää huomioida rahtia lisätessä tai poistaessa.
 - b. **Autolautta:** uusina attribuutteina kapasiteetti (*final int*), matkustajamäärä (*int*) ja ajoneuvojen määrä(*int*). Yksi ajoneuvo vie tilaa 10 yksikköä ja yksi matkustaja yhden yksikön; kapasiteettia ei voi ylittää. (20 p.)
2. Jatketaan viime demojen Joukkue- ja Pelaaja-luokkien kehittämistä. Toteuta seuraavat luokat:
 - a. **Jalkapallojoukkue:** perii Joukkue-luokan. Pelaajien maksimimäärä on 22 (2 maalivahtia, 20 kenttäpelaajaa). Lisäksi Jalkapallojoukkueella pitää olla tieto sen kotistadionista.
 - b. **Jalkapallopelaaja:** Perii Pelaaja-luokan. Sisältää uusina attribuutteina tiedon siitä, onko pelaaja vasen- vai oikeajalkainen ja pelaajan viikkopalkan. Kirjoita attribuuteille lisäksi asetus- ja havainnointimetodit. Huomaa, että jalkapallojoukkuetta tulostaessa tulee tulostaa Pelaaja-luokan attributtien lisäksi myös Jalkapallopelaaja-luokan attribuuttien tiedot.(20 p.)
VIHJE: Joukkue- ja Pelaaja-luokkiin ei tarvita muutoksia.
3. Aloitetaan viime demoissa esitetyn kurssienhallintasovelluksen toteutus. Toteuta ainakin seuraavat luokat toiminallisuuksineen:
 - a. **Opiskelija, Opettaja** (yhteinen yliluokka **Yliopistolainen**)
 - b. **Kurssi**
 - c. **Periodi** (sisältää kaikki yhden periodin kurssit)
 - d. **Lukuvuosi** (Sisältää 5 periodia)(30 p.)
4. Jatketaan linketety listan toteutusta. Muokataan toteutusta niin, että
 - a. listasta tulee **kaksisuuntainen**. Jokainen alkio tietää siis seuraavan ja edellisen alkion listassa.
 - b. listasta tulee **geneerinen**. Lista voi siis tallentaa minkä tyyppistä tietoa vain (kuten esimerkiksi ArrayListiin tai HashMapiin).

Muista päivittää lisäys- ja poistometodien toiminnallisuus. (30 p.)

LISÄTEHTÄVÄ: Tehtävä liittyy Moodlesta löytyvään Verkkokauppa-esimerkkiin. Keksikää demoissa mahdollisimman monta tapaa, jolla verkkokaupasta voi varastaa puhelimen muuttamalla vain Ostotapahtuma-luokan `suoritaPuhelimenOstotapahtuma`-metodia. Varastaminen on siis suoritettu onnistuneesti, jos metodin suorituksen jälkeen varkaan omaPuhelin-attribuutin arvo != null ja varallisuus-attribuutin arvo kasvaa tai pysyy samana.