

Olio-ohjelmoinnin perusteet – Demo 1

1. Toteuta luokka Laiva. Luokan tulee sisältää ainakin attrivuutit nimi, pituus, syväys ja nopeus solmuina. Toteuta lisäksi metodi, jolla voidaan laskea kuinka nopeasti laiva kulkee annetun matkan. (20 p.)
2. Toteuta luokat Pelaaja ja Joukkue. Pelaajalla tulee olla attribuutit nimi, numero ja pelipaikka (hyökkääjä, puolustaja tai maalivahti). Joukkue sisältää n pelaajaa ja sen tulee sisältää toiminnallisuudet pelaajien lisäämiseksi tai poistamiseksi joukkueesta. Lisäksi Joukkue-luokan tulee sisältää metodi, jonka avulla voidaan tulostaa ruudulle kaikki joukkueen pelaajat tietoineen. (20 p.)
3. Suunnittele yliopiston kurssienhallintaa hoitava sovellus. Pohdi, **mitä** luokkia (esim. opiskelija) kurssienhallintaa varten tarvitaan. Mieti myös mitä tietosisältöä ja toiminnallisuutta ne sisältävät. **Miksi** piirre on valittu luokkaan ilmaistaan sitten esityksessä. Luokkien toiminnallisuutta ei tarvitse toteuttaa, pelkkä toiminnallisuuksien listaus riittää. Jos UML-mallinnuskieli (tulee kurssin loppupuolella) on jo tuttu, voi mallinnuksen suorittaa sillä. Jos ei, esimerkiksi JavaDoc-tyyppinen piirteiden listaus kommentteineen+alkuehtoineen on riittävä. Esimerkki:
Avaruusalus:
//nopeampi kuin warp 10 on mahdotonta (paitsi, kun elokuvan juoni toisin vaatii)
private static int WARPMAX=10
private String nimi
// nopeus hyppykertoimena; valon nopuden kerroin
private int maxnopeus

/ GETTERIT JA SETTERIT*/*
public String haeNimi()
public String haeNopeus()

*/*Lähetää aluksen matkalle*
**@param x != null*
**/*
public void matkustaPlaneetalle(Planeetta x)

/ Käynnistää valoa nopeamman hypyn. Alcubierren warp drive mahdollistaa teoreettisesti valoa*
** nopeamman liikkumisen paikallisessa aika-avaruudessa.*
** @param warpKerroin > 0*
**/*
private void käynnistäAlcubierreMoottori(int warpKerroin)
(30 p.)
4. Linkitetty lista on tietorakenne, jossa jokainen alkio tietää alkion, joka tulee listassa seuraavaksi. Toteuta yksisuuntainen linkitetty lista (eli listan alkio tietää seuraavan alkion, mutta ei edellistä), johon voi tallentaa kokonaislukuja. Toteuttaa tulee siis alkion lisäys-, poisto- ja hakumetodit. (30 p.)