

Demo 8 / 14.3

Huomatkaa poikkeava päivä ja se että tällä kertaa 14.3 pidetään demojen palautustilaisuus vain klo 16:15-18.

Tästä alkaa uusi demojakso ja nyt laskennallinen maksimi joka kerta on 8 tehtävää. Enää ei siis ole mahdollista kerätä yli 100% (eikä tarvitse :-). Ja ComTest tai JUnit testit joka tehtävään, joihin se on mielekkäästi mahdollista (merkitty (T)).

- 1*2. Täydennä edellisen demon tehtävän 7.7-8 vastaus [MakiHyppy.java](#) kunnon olio-ohjelmaksi niin, että seuraava testiohjelma toimii (huom! kaikkien "ominaisuuksien" ei tarvitse olla attribuutteina) (ks. [Java-kuva](#))

```
public void kisa() {
    Kilpailija toni = new Kilpailija("Toni", 3);
    Kilpailija matti = new Kilpailija("Matti", 7);
    toni.tulosta();
    matti.tulosta();

    toni.setPituus(1, 107);
    toni.setPituus(2, 100);
    toni.setTuomari(2, 1, 19.0);
    toni.setTuomari(2, 2, 18.0);
    toni.setTuomari(2, 3, 19.5);
    toni.setTuomari(2, 4, 18.0);
    toni.setTuomari(2, 5, 20.0);

    matti.setPituus(1, 125);
    matti.setTuomari(1, 1, 20.0);
    matti.setTuomari(1, 2, 20.0);
    matti.setTuomari(1, 3, 20.0);
    matti.setTuomari(1, 4, 20.0);
    matti.setPituus(2, 109);
    matti.setTuomari(2, 1, 20.0);
    matti.setTuomari(2, 2, 20.0);
    matti.setTuomari(2, 3, 20.0);
    matti.setTuomari(2, 4, 20.0);

    toni.tulosta();
    matti.tulosta();
}

public static void main(String[] args) {
    Makihyppy kisa = new Makihyppy();
    kisa.kisa();
}
```

- 3*. Muuta demotehtävän 7.4 vastaus [Kirje.java](#) sellaiseksi, että kahden taulukon sijasta onkin vain yksi taulukko, jonka alkioina on Hinta -luokka (ominaisuudet hinta ja paino). Kirjoita myös lisäksi funktio `double postimaksu(int paino)` .(T)

Toteuta vielä siten, että on yksi 2-ulotteinen taulukko (n x 2), jonka 1. sarakkeessa (sarake 0)

on paino reaalitylukuna ja toisessa sarakkeessa (sarake 1) hinta reaalitylukuna. (T)

4*5. Katso esimerkkiä [Vaihtoehdot2.java](#) ja [AanestysLiittyma.java](#). Noissa on toisessa tiedostossa kaikki ”logiikka” (eli tietorakenteen käsittely) ja toisessa kaikki komentorivikäyttöliittymään kuuluva asia. Demo 7 mallivastauksessa [AstiaPeli.java](#) on kuitenkin sekä käyttöliittymää, että logiikkaa samassa tiedostossa. Erotta tästä vastaavalla tavalla itse AstiaPeli ja KomentoriviAstiaPeli. Ota [demo 7 vastauksista](#) muutkin tarvittavat tiedostot jotta saat pelin ensin käännettyä. Koita saada niin, että pääohjelma on täsmälleen samanlainen kuin GraafinenAstiaPeli-luokassa. Tarvittaessa kirjoita muutama pieni metodi lisää luokkaan AstiaPeli. Testit Bonus-tehtävänä.

6*. Kirjoita aliohjelma matriisinSuurin, joka palauttaa 2-ulotteisen matriisin suurimman alkion. (T)

7. Edellisen kerran tehtävän 7.8 vastauksen [Rajat.java](#) funktiota summa käyttäen kirjoita aliohjelma matriisinSumma, joka laskee 2-ulotteisen reaalitylukumatriisin summan (älä kopioi, vaan kutsu funktiota, muista että matriisi on taulukko riveistä!). (T)

8. Ota selvää miten päivämäärä saadaan selville Javan kirjastokutsuilla. Kirjoita metodi paivays, jolla saadaan nykyinen päiväys selville [Pvm.java](#):n Pvm tyyppiseen olioon. Muuta tarvittaessa myös luokan muodostaja ja alusta -metodi sellaiseksi, että jos alustuksessa ei anneta kaikkia arvoja, arvona käytetään nykypäiväystä puuttuville arvoille (pois saa jättää oikealta): (T, pitää muuttaa Pvm2Test -luokkaa hieman) Esimerkiksi:

```
Pvm tammi2005 = new Pvm(1,1,2005)=> 1.1.2005
Pvm tammi2011 = new Pvm(1,1); // => 1.1.2011
Pvm maaliskuu2011 = new Pvm(1); // => 1.3.2011
Pvm tanaan = new Pvm(); // => 13.3.2011
// (testattu 14.3.2011)
```

B1. Malliksi toisenlainen [GraafinenAstia](#). Huonoa tuossa on vielä montakin asiaa. Yksi on kuitenkin se, että astian koodiin on sotkettu paljon grafiikkaa, jolla voisi olla muutakin käyttöä. Ota grafiikkaan liittyvä koodi ja tee siitä Pylvas -luokka, joka voidaan testata seuraavalla pääohjelmalla:

```
public static void main(String[] args) {
    Pylvas p5 = new Pylvas(1,5,0);
    Pylvas p8 = new Pylvas(1,8,2);
    // muodostajalle pylvään leveys, korkeus ja väritetyn osan määrä
    // väritetyn osan korkeudesta käytetään nimeä väli

    Window window = new Window();
    window.scale(0,0,5,8);
    p8.move(2,0,0);
    p8.setValiColor(Color.RED);

    window.add(p8);
    window.add(p5);
    window.showWindow();

    Syotto.kysy("Lisää 5 pylvään väliä");           p5.setVali(4);
    Syotto.kysy("Pienennä 8 pylvään korkeutta");    p8.setKorkeus(6);
}
```

```
}
```

B2. Muuta KomentoriviAstiaPeli-luokan testit taas toimiviksi.

G1-4. Malliohjelman [vaiheen 5](#) luokassa `Jasenet` ([Jasenet.java](#)) on metodi `anna` joka palauttaa `Jasen` -luokan viitteen ([Jasen.java](#)). Tässä on se vika, että luokassa `KerhoSwing` ([KerhoSwing.java](#)) olevassa metodissa `tulostaValitut` voitaisiin muuttaa `Jasen` -luokan olion arvoa (ks. myös [Kerho.java](#)):

```
431     public void tulostaValitut(JTextArea text) {
432         PrintStream os =
433             TextAreaOutputStream.getTextPrintStream(text);
434         os.println("Tulostetaan kaikki jäsenet");
435         for (int i = 0; i < kerho.getJasenia(); i++) {
436             Jasen jason = kerho.annaJasen(i);
437             jason.vastaa_aku_ankka(); // esim. näin muutettaisiin
438             tulosta(os, jason);
439         }
440     }
```

Mieti ja toteuta millainen pitäisi olla metodin `anna` paluarvo, jotta metodia käyttävät eivät voisi mitenkään "pilata" palautetun jäsenen arvoa. Vinkki: muista rajapinnat.