



Luento 4

ITKP102 Ohjelmointi 1

Antti-Jussi Lakanen

Tarkennuksia ja kertausta



Tarkennuksia ja kertausta

- Pisteet päivittyvät Omat tiedot –sivulle automaattisten arvioiden sekä itse syötettyjen pisteiden perusteella



Pari poimintaa AMA-palstalta (1)

- Ks. TIM-Kotisivu -> Oppimateriaali -> Kysymyksiä ja vastauksia
- **Riittääkö tämän kurssin materiaali opiskeluun tällä kurssilla?**
 - Käytössä luentomoniste, sekä tietenkin pidetyt luennot. Lisäksi jotakin täydentävää materiaalia on ”Lisämateriaali”-kohdassa.
 - Jos jotakin mielestäsi puuttuu, anna palautetta



Pari poimintaa AMA-palstalta (2)

- Mihin asti tämän kurssin osaaminen riittää? Voinko tulkita koodia asiakasympäristöissä?
 - Riippuu tavoitteistasi
 - Jos tämä on ainoa pakollinen ohjelmointikurssisi, ja tavoitteenasi on suuntautua teknisempiin hommiin, niin ilman muuta suosittelen ottamaan jatkokursseja.
 - Teknisen ja toiminnallisen kokonaiskuvan tulkitseminen asiakasjärjestelmästä ei onnistu vielä tämän kurssin tiedoilla.



Työkaluista

- Tästä eteenpäin:
- .NET 7
- Rider tai Visual Studio (Visual Studio Code ei vielä täysin käyttökelpoinen Jypeli-projekteille)
- Työskentely Riderissa, ratkaisut copy-pastetaan TIMiin

Muuttujat



Mikä on muuttuja (1)

- Muistista lohkaistu säilytyspaikka tiedolle. Tälle säilytyspaikalle on annettava nimi.
- Muuttujaan voidaan tallentaa tietoa suorituksen ajaksi
- Muuttujan sisältämä arvo voidaan lukea
- Muuttujan tietoa voidaan muuttaa
- Vahvasti (ja staattisesti) tyypitetyissä kielissä muuttujalla on aina (jokin tietty) tyyppi



Mikä on muuttuja (2)

- Lähdetään liikkeelle muuttujan määrittelystä
- [tyyppi] [nimi];
- Näin kirjoitettuna syntyy muuttuja jolla
 - ei ole arvoa, tai
 - jonka ”olematonta” arvoa ei voi käyttää



Sijoitus (1)

- Arvon asettaminen tapahtuu sijoittamalla käyttäen `=`-operaattoria (kutsutaan sijoitusoperaattoriksi)
- Muuttujaan voi sijoittaa arvon määrittelyn yhteydessä...
 - `[tyyppi] [nimi] = [arvo];`
- ja/tai myöhemmin
 - `[nimi] = [uusiArvo];`
- Huomaa, että tyyppi mainitaan vain määrittelyn yhteydessä



Sijoitus (2)

- Sijoituksessa suunta on oikealta vasemmalle
- Muuttuja voi sisältää kerrallaan vain yhden arvon
- Tässä mielessä muuttuja eroaa huomattavasti matemaattisen muuttujan käsitteestä



Nimi ja arvo

- Nimi
 - Tulee kuvailla sisällön merkitystä
 - Helpottaa koodin lukemista
 - Ei voi olla C#:n avainsana (engl. keyword), kuten `int`, `double`, `public`
- Arvo
 - Mikä tahansa kyseisen tyypin sallittu lauseke



Perustietotyypit (1)

- Lukutyypit
 - `int`, kokonaisluku (32 bittiä)
 - `double`, desimaaliluku (liukuluku, 64 bittiä)
 - `float`, desimaaliluku (liukuluku 32 bittiä)
 - Nämä riittävät lukutyypeistä tälle kurssille. Lisäksi on `short`, `uint`, jne.



Perustietotyypit (2)

- Totuusarvo
 - `bool` (1 bitti)
- Merkki
 - `char` (16 bittiä)
- Merkkijono
 - `String`
 - Demoissa tarvitaan: `string syote = Console.ReadLine();`
- Ks. [moniste](#)



Muuttujan arvon muuttaminen

- Muuttujaan voidaan siis sijoittaa uusi arvo.
- Sitä ei tarvitse sitoa mihinkään tiettyyn arvoon.
- `int lapsenlka = 3;`
- `lapsenlka = 4;`



Muuttujan arvon muuttaminen

- Muuttujaan voidaan myös sijoittaa toisen muuttujan sisältämä arvo
- `int isoveljenIka = lapsenIka + lapsenIka;`



Vakion arvoa ei voi muuttaa!

- `const int SORMIA = 2;`
- `const`-avainsana estää `SORMIA`-muuttujan arvon muuttamisen
- `SORMIA = 3; // käännösvirhe`
- Huomaa vakion kirjoitusasu.



Huomautus: arvojen vaihtaminen

- `int eka = 1;`
- `int toka = 2;`
- `int temp; // tilapäismuuttuja`
- `temp = toka;`
- `toka = eka;`
- `eka = temp;`
- Mikä on `toka`-muuttujan arvo nyt?



Tietotyypeistä

- Edellä mainittuja tietotyypppejä kutsutaan alkeistietotyypeiksi, ne ovat kieleen sisäänrakennettuja perustyypppejä
- Voimme tehdä myös omia tietotyypppejä
- Jypelissä mm.
 - / `PhysicsObject` ja `GameObject`
 - / `Vector` (2d-vektori)
 - / ...ja paljon muita.