



# Luento 9

ITKP102 Ohjelmointi 1

Antti-Jussi Lakanen



# Vielä pari silmukka-esimerkkiä

- while-silmukassa toistoehto tarkastetaan `_ennen_` silmukan runko-osan suorittamista
- do-while-silmukassa toistoehto tarkastetaan silmukan runko-osan suorittamisen jälkeen
- while: sopii erityisen hyvin tilanteisiin, joissa toistojen määrä ei ole ennalta tiukasti kiinnitetty
- do-while: Sopii erityisen hyvin tilanteisiin, joissa tehdään aina vähintään yksi toisto



# Esimerkit Riderissa

# Debuggaus



# Virheenjäljitys eli debuggaus

- ”Bugi” on ilmiö tai virhe joka estää ohjelmaa toimimasta halutulla tavalla
- Debuggaamisella tarkoitetaan ohjelmassa ilmenneen bugin jäljitystä ja korjaamista (”de-bugging”)



# Virheenjäljitys eli debuggaus (jatkuu...)

- Debuggaukseen voidaan lukea erilaisia taktiikoita. Muun muassa:
  - Ohjelman tilan tutkiminen interaktiivisesti (engl. interactive debugging)
  - Printtaus (engl. print debugging / tracing)
  - Post-mortem, esim. kutsupinojen ja muistidumppien tulkinta
- Kaikilla näillä on sama tavoite, eli korjata rikkinäinen ohjelma.
- Ohjelmointi 1:ssä tehdään ensisijaisesti näistä ensimmäistä, joskus myös toista



# Debuggaus Riderissa

- Keskeytyskohta (break point)
- Askellus (step into, step over)
- Paikallisten muuttujien tila (local variables) ja niiden muuttaminen
- Seurattavat (Watch)
- Kutsupino
- Ehdollinen keskeytyskohta

# Taulukot





# Taulukot

- Saman tyyppisiä arvoja voidaan koota yhteen tietorakenteeseen
- Yksinkertaisin tietorakenne C#:ssa on taulukko
- Esimerkiksi asunnossa asuvien henkilöiden nimet:
  - `string[] henkilot = {"Pertti", "Irmeli", "Kalle"};`
- Tai tilien saldot
  - `double[] tilienSaldot = {150.81, 0.05, 123.45, -137.42};`



# Taulukot

- Taulukon arvoja kutsutaan alkioiksi tai elementeiksi
- Elementteihin päästään käsiksi taulukkomuuttujan ja hakasulkujen avulla
- `string[] henkilot = {"Pertti", "Irmeli", "Kalle"};`
- Esimerkiksi
  - `Console.WriteLine(henkilot[0]);` tulostaisi Pertti,
  - `Console.WriteLine(henkilot[1]);` tulostaisi Irmeli



# Taulukon luominen

- Taulukon pituutta ei voi sen luomisen jälkeen muuttaa
- Taulukon voi luoda kahdella tavalla
  1. `Tyyppi[] nimi = new Tyyppi[koko];`
    - Tällöin taulukon sisältö on asetettava kuhunkin alkioon erikseen
  2. Taulukon sisältö voidaan myös sijoittaa luomisen yhteydessä
    - `Tyyppi[] nimi = new Tyyppi[] {alkio1, alkio2, ...};`
    - Huomaa, että kokoa ei tarvitse tällöin erikseen asettaa



# Taulukon luominen

- Joillain tyypeillä, kuten int, double, string, toimii myös hieman lyhyempi alustus ilman new Tyyppi[] -osaa.



# Tähän jäätiin luennolla 9...



# Taulukot

- Tavallisesti taulukon alkiot liittyvät jollain tavalla loogisesti toisiinsa