

# Johdatus olio- ohjelmointiin

Antti-Jussi Lakanen  
Ohjelmointi 1, C# /  
kevät 2011



# Yleistä olioista ja OOP:sta

- OOP, eng. Object Oriented Programming
- C#:ssa *oliot* luodaan käyttämällä (olio)*luokkia*
- Olio tarkoittaa yksinkertaisesti ohjelman osasta
- *Olio-ohjelmointi* on sitä, että
  - ... ohjelma muodostuu kokoelmasta yhteistyössä toimivia oliota
  - Jokainen olio voidaan nähdä itsenäisenä pienenä koneena, jolla on tietty rooli tai vastuu.

# Yleistä olioista ja OOP:sta (jatkuu)

- Jokainen olio kuuluu johonkin joukkoon (luokkaan)
- Olioiden yhteisten ominaisuuksien perusteella niistä voidaan kirjoittaa yleinen määritelmä (edelleen luokka)
- Oliot ovat asioita, jotka viittaavat tosielämän asioihin tai toimintoihin (koira, päivämäärä, aurinkoinen päivä,...)
- Jokainen olio on yksilöllinen ja jokaisella oliolla on...
  - Attribuutit (ominaisuudet, esim. koko, ikä, paino)
  - Toiminta (mitä tekee, esim. lukee, opiskelee, nukkuu)

# Esimerkki: eläinlääkärin toimisto

- Millaisia ”objekteja” tai asioita eläinlääkärissä on? (substantiivit)
  - Asiakas
  - Eläin (monenlaisia eläimiä)
- Kaikki nämä vieraat eivät ole samanlaisia
  - Pitää pystyä erottelemaan ne asettamalla niille **ominaisuuksia**



# Esimerkki (jatkuu)

- Eläimen ominaisuuksia: rotu, nimi, ikä, paino...
- Asiakkaan ominaisuuksia: nimi, osoite...



# Esimerkki (jatkuu)

- Mitä eläinlääkärissä tehdään eläimille? (verbit)
  - Esimerkiksi rokotetaan, syötetään, punnitaan, jne.
  - Toimintoja, joita eläimelle tehdään eläinlääkärissä
- Entä asiakas, mitä hän tekee?
  - Ainakin maksaa laskun
- Näistä verbeistä tulee ohjelmakoodin **aliohjelmia**
- Aliohjelmia voidaan ”kutsua”, esimerkiksi

SyotaElainta (elain1, 2)

Eläimen  
”tunniste”

Paljonko ruokaa  
Annetaan (esim. kg)

# Esimerkki (jatkuu)

- Jokaisella eläimellä on samat ominaisuudet: rotu, nimi, ikä, paino
  - Eläin a: Kissa, Pörhö, 3-v, 2 kg
  - Eläin b: Koira, Roope, 4-v, 15 kg
  - Eläin c: Tarantella, Lurska, 6 kk, 200 g
- Nämä eri eläimet ovat ohjelmointikielen **olioita**, jotka ovat peräisin **eläin-luokasta**
  - Luokka on tavallaan piparkakkumuotti, jolla tehdään useita (samanmuotoisia) piparkakkuja, siis olioita
- Samoin kullakin asiakkaalla vastaavasti yksilölliset tiedot

# Esimerkkiluokka: Koira

- Mitä ominaisuuksia koiralla on? (Huom! Unohda vielä ohjelmakoodi!)
  - Ainakin nimi, ikä, rotu
  - Ehkäpä ”onko koira nälkäinen” voisi olla hyvä tieto
  - Paino myös
- Mitä koiralla/koiralle tehdään?
  - Käytetään pihalla
  - Punnitaan
  - Tiedustellaan onko nälkäinen (ehkä koira itsekin sen ilmoittaa...)
  - Tietenkin syötetään



# Esimerkkiluokka: Koira

- Koira (luokka)
  - Ominaisuudet
    - rotu : String
    - paino : double
    - nimi : String
    - syntymavuosi : int
    - nalkainen : bool
  - Toiminnot
    - Syota (maara : double) : void
    - Punnitse() : double
    - OnkoNalkainen() : bool
    - OvatkoElintoiminnotKunnossa() : bool

# Esimerkkiluokka: Koira

- Kirjoitetaan malliohjelma...

# Olioista

- Luokka on siis kokoelma tietoja, joilla määritellään olion attribuutit (ominaisuudet) ja toiminta

# Koira-luokka: ongelma

- Jos olisi kyseessä eläinlääkärin toimisto, niin luultavasti meillä olisi muitakin eläimiä kuin koiria
- Pitäisikö meillä olla myös luokat
  - Kissa
  - Kaarme
  - Tarantella
  - Hevonen
  - Marsu
  - Gerbiili
  - ...

# Koira-luokka: ongelma (jatkuu)

- Tässä tuskin olisi järkeä!
- Suunnitellaanpa luokka uudestaan sopivaksi eläinlääkärille (yleistetään!)
  - Huom! Esimerkki on kuvitteellinen eikä välttämättä vastaa tosielämän vaatimuksia.

# Esimerkkiluokka: Elain

- Mitä ominaisuuksia eläimellä on?
  - Laji (koira, kissa, marsu...)
  - Nimi, ikä, ehkäpä rotu
  - Kaikki eläimet syövät, joten siitä jotain tietoa
  - Paino myös tarvitaan
- Mitä eläimelle tehdään? No ainakin...
  - Jollain tavalla aktivoidaan / ulkoilutetaan / tms...
  - Punnitaan tai tarkistetaan elintoimintoja
  - Tiedustellaan onko nälkäinen
  - Tietenkin syötetään

# Esimerkkiluokka: Elain

- Elain (luokka)
  - Ominaisuudet
    - laji : String
    - rotu : String
    - paino : double
    - nimi : String
    - syntymavuosi : int
    - nalkainen : bool
  - Toiminnot
    - Ulkoiluta() : void
    - Syota(maara : double) : void
    - Punnitse() : double
    - OnkoNalkainen() : bool
    - OvatkoElintoiminnotKunnossa() : bool