Ohjelmoinnin peruskurssi Y1

CS-A1111

Olio-ohjelmointi

Oppimistavoitteet: tämän videon jälkeen

- Tiedät, mitä *oliot* ovat ja miksi niitä käytetään ohjelmoinnissa.
- Tiedät, miten olioiden ominaisuuksia ja niille mahdollisia toimenpiteitä voi määritellä *luokan* avulla.
- Tiedät, miten olioita voi luoda.

Ongelma

- Esimerkki: halutaan laatia ohjelma, joka käsittelee erään ohjelmointikurssin opiskelijoita. Kurssilla on noin 1000 opiskelijaa.
- Jokaisesta opiskelijasta halutaan ohjelman käyttöön ainakin nimi, opiskelijanumero, tenttiarvosana ja harjoitusarvosana.
- Ongelma: miten opiskelijoiden tietoja esitetään ja käsitellään ohjelmassa?

Ongelman huonoja ratkaisuja

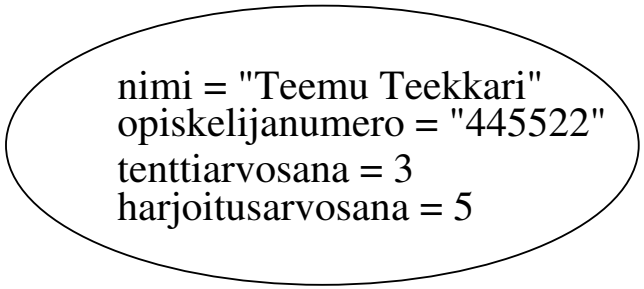
- 1. ratkaisu (surkea): otetaan käyttöön 4000 muuttujaa eri arvoja varten.
- 2. ratkaisu (huono): otetaan käyttöön 4 eri listaa: `nimet`, `opiskelijanumerot`, `tenttiarvosanat` ja `harjoitusarvosanat`. Jokaisessa listassa on 1000 alkiota ja saman opiskelijan tiedot ovat listassa aina samalla indeksillä.
- 3. ratkaisu (parempi): tehdään yhden opiskelijan tiedoista lista, jossa on neljä alkiota. Kurssin kaikista opiskelijoista muodostetaan lista, jonka alkiot ovat listoja.
- 4. ratkaisu (parempi): tehdään yhden opiskelijan tiedoista lista, jossa on kolme alkiota, ja tallennetaan nämä sanakirjaan niin, että avaimena on opiskelijan opiskelijanumero.

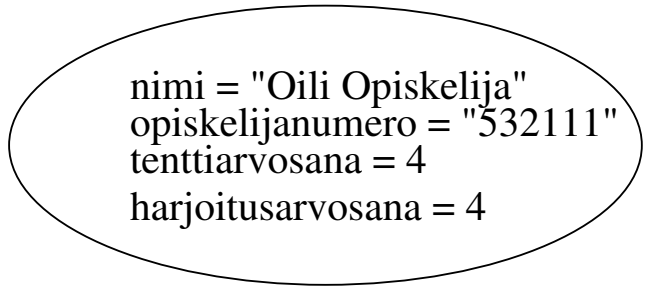
Ratkaisu olioiden avulla

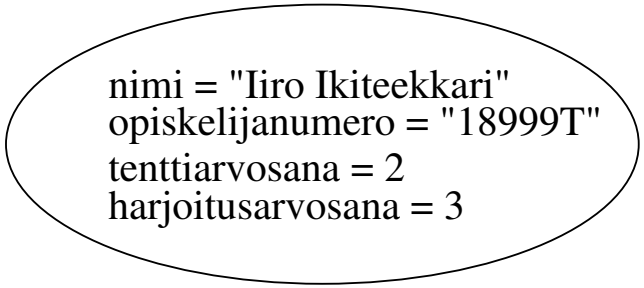
- Olioita käyttävä ratkaisu:
  - Tehdään jokaista oikeaa opiskelijaa kohti ohjelmaan yksi `Opiskelija`-olio.
  - Luokassa `Opiskelija` kerrotaan, millaisia `Opiskelija`-oliot ovat ja mitä toimintoja niille voi tehdä.
  - Kurssin kaikkia opiskelijoita esitetään `Opiskelija`-olioita sisältävänä listana.
- Olioita käyttävän ratkaisun etuja:
  - Yhden opiskelijan tiedot muodostavat yhden kokonaisuuden.
  - Opiskelijan eri tiedot voidaan nimetä selvästi.
  - Voidaan määritellä myös oliolle mahdolliset toimenpiteet.

Mitä oliot ovat?

- Olioihin säilötään tietoa olion ominaisuuksista ja oliot osaavat “tehdä temppuja”
- Jokaisella oliolla on samat kentät (ominaisuudet), mutta niissä omat arvot, esimerkiksi `Opiskelija`-oliolla kentät `nimi`, `opiskelijanumero`, `harjoitusarvosana` ja `tenttiarvosana`.







Luokka ja olio

- Luokassa määritellään, millaisia luokan oliot ovat ja mitä operaatoita olioille voidaan tehdä.
- `Opiskelija`-olioiden käsittelemiseksi kirjoitetaan luokka `Opiskelija`.
- Luokkaan kirjoitetaan metodit, jotka määrittelevät `Opiskelija`-olioille mahdolliset operaatiot.
- Luokan määrittely aloitetaan luokan otsikolla:

```
class Opiskelija:
```

Olioiden luonti

- uusia `Opiskelija`-olioita voidaan luoda antaen sille alkuarvot:

```
kurssilainen1 = Opiskelija("Matti Virta", "212233")
kurssilainen2 = Opiskelija("Oili Lahti", "574455")
```

- `Opiskelija`-luokassa metodi `__init__` määrittelee, mitä tapahtuu, kun luodaan uusi luokan olio, ja mitä alkuarvoja tarvitaan tällöin.

```
def __init__(self, annettu_nimi, numero):
    self.__nimi = annettu_nimi
    self.__opiskelijanumero = numero
    self.__tenttiarvosana = 0
    self.__harjoitusarvosana = 0
```

Olioiden luonti, esimerkki

```
kurssilainen1 = Opiskelija("Matti Virta", "212233")
kurssilainen2 = Opiskelija("Oili Lahti", "574455")
```