

# Ohjelmoinnin perusteet

## R0027

---

5 op



# Agenda

---

- **Kurssin esittely**
- **Johdatus ohjelmointiin ja Java-ohjelmointikieleen**
- **Viope**
- **Kehittämisharjoitus**



# Esitiedot

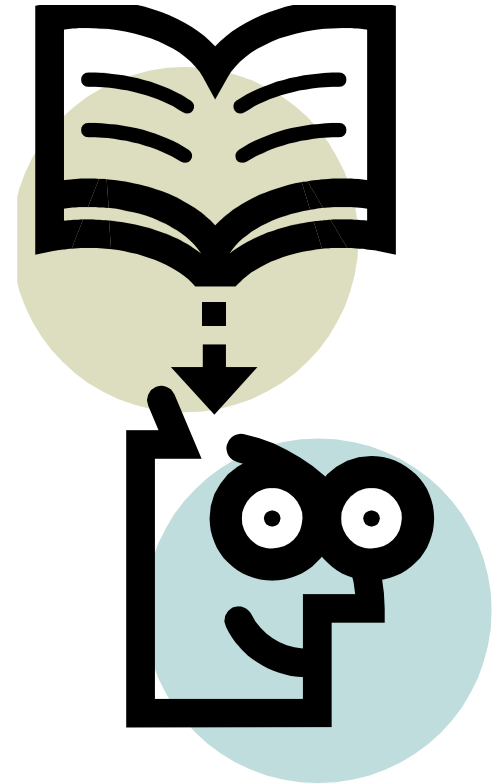
---

- Tietokoneen peruskäyttö
  - Mikä on tiedosto?
  - Mikä on kansio/hakemisto?
- Webin käyttö
- Englannin kielen tyydyttävä osaaminen
- **Kurssilla ei edellytetä esitietoja ohjelmoinnista 😊**

# Sisältö

---

- ❑ Ohjelmoinnin perusteet
- ❑ Java-ohjelmointikieli
- ❑ Eclipse-kehitysympäristö



# Lähiopetusta, harjoituksia ja kotitehtäviä

---

- ❑ **Teoriatunnit**
- ❑ **Kehittämisharjoitukset**
- ❑ **Viope-kotitehtävät**

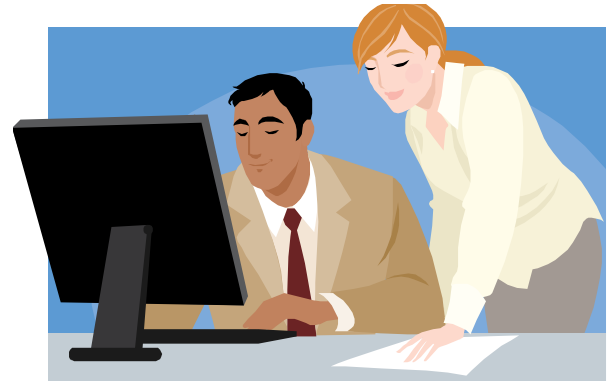


*Joka viikko opitaan uutta asiaa, jonka hahmottaminen edellyttää edellisten viikkojen asioiden ymmärtämistä.*

# Kehittämisharjoitukset

---

- Harjoituksina tenttikysymysten kaltaisia tehtäviä

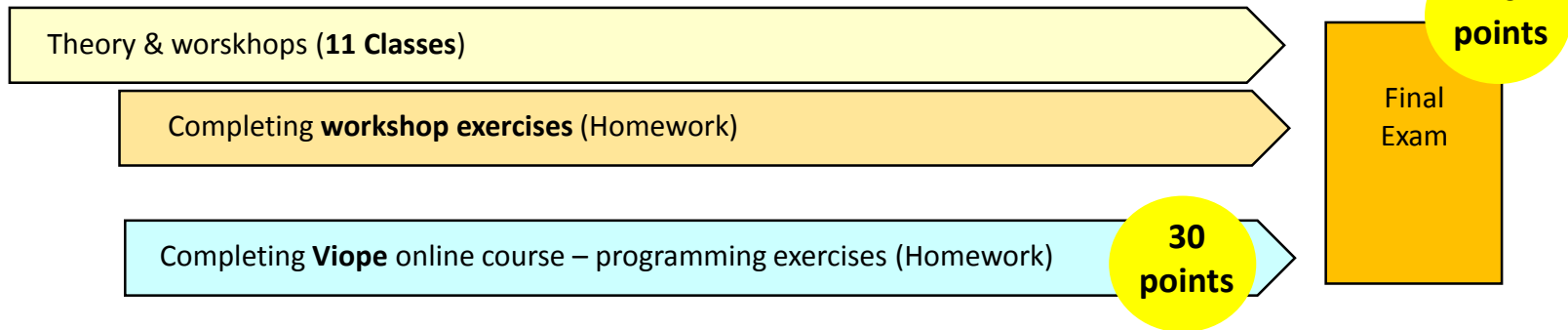


# Viope-kotitehtävät

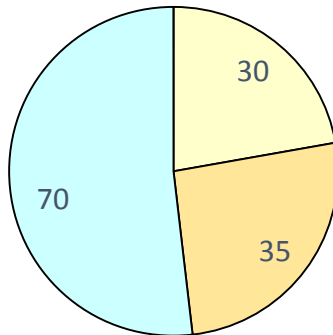
---

- *Virtuaalinen oppimisympäristö ohjelmoinnin opiskeluun*
  - *Teoriaa*
  - *Monivalintatehtäviä*
  - *Ohjelmointitehtäviä*
- <https://vw4.viope.com/login?org=laurea>
- Ilmoittaudu kurssille tänään  
***Java perusteet (R0027 Kevät 17)***

# Työmäärä ja arviointi



## Workload – 5 credits (135 hours)



□ Classes □ Workshop (homework) □ Viope (homework)

## EVALUATION:

- 1) Completing Viope's programming exercises: 0-30 points
- 2) Final exam: 0-70 points

FINAL POINTS: Sum of the above (0-100 points)

Final grade in the course:

- |             |                                       |
|-------------|---------------------------------------|
| 0-40 pts:   | Grade of "0" (not passing the course) |
| 41-48 pts:  | Grade "1" (satisfactory)              |
| 49-59 pts:  | Grade "2" (satisfactory)              |
| 60-71 pts:  | Grade "3" (good)                      |
| 72-84 pts:  | Grade "4" (very good)                 |
| 85-100 pts: | Grade "5" (excellent)                 |



# Halki suuren urakan...

---

- Kurssia ennen
  - Sinulla on jotain odotuksia ja mielikuvia ohjelmoinnista
  - Asetat itsellesi korkeita oppimistavoitteita
- Kurssin aikana
  - Yllätyt työmäärästä
  - Et luovuta, vaikka asia tuntuu välillä vaikealta
  - Jos tiput kärryiltä, kirit kiinni ajoissa
- Kurssin jälkeen
  - Saat palkinnoksi taidon tehdä ohjelmia
  - Huomaat, että olet oppinut todella paljon uutta
  - Mahdollisesti jopa pidät "koodaamisesta"

# Työtä on siis paljon

---

- ❑ Kehittämistunneilla hommia saa tehdä pareittain, sopivin väliajoin koodaajaa vaihdellen
  - Kommunikointi auttaa usein ajatusten jäsentämisessä
  - Kaksinkertainen mahdollisuus havaita virheet
- ❑ Yksinkin tekeminen on mahdollista
- ❑ Kotitehtäviin saa kysyä apua, mutta niitä ei saa plagioida suoraan
- ❑ **Tällä kurssilla opetellaan ratkaisemaan ongelmia, joten ole kärsivällinen**
- ❑ Itsekuri estää tenttipaniikin

# Aikataulu

---

- Optimasta (muutokset mahdollisia)

# Kurssimateriaali

---

## □ Ensisijaiset lähteet

- Kalvoesitykset
- Kehittämisharjoitukset (Workshop exercises)
- Viope
- **Kirja: Java 2 – Ohjelmoinnin peruskirja**  
**Kosonen, P., Peltomäki, J., Silander, S.**  
2008, 4. laitos, 3. painos (myös vanhemmat painokset käyvät)

# Kurssimateriaali

---

## □ Toissijaiset lähteet

### ■ Java Tutorial

<http://download.oracle.com/javase/tutorial/>

### ■ Java API

<http://download.oracle.com/javase/6/docs/api/>

### ■ Eclipse Help

<http://help.eclipse.org>

### ■ Google

<http://www.google.fi>

### ■ Youtube

<http://www.youtube.com>

# A

# Johdatus ohjelmointiin ja Java-ohjelmointikieleen



## Viopen luvut 1&3

# Ohjelmointi

---

- ❑ **Ohjelmointi on ohjelmien tekemistä**
- ❑ Mikä on ohjelma?
- ❑ Miten ohjelma toimii?
- ❑ Mikä on ohjelmointikieli?

# Ohjelmat

---

- Ohjelmat (*programs*) ohjaavat laitteiden (tietokoneiden) toimintaa
  - Mobiililaitteet
  - Yleiskäyttöiset tietokoneet
  - Sulautetut järjestelmät (*embedded systems*), esim. pankkiautomaatit ja pesukoneet



# Ohjelmatyyppejä

---

- ❑ Työpöytäsovellukset
  - Calculator
  - Media Player
- ❑ Työvälineohjelmat
  - MS Word
  - Adobe Photoshop
- ❑ Verkkosovellukset
  - Optima
  - Verkkokauppa
- ❑ Mobiilisovellukset
- ❑ Käyttöjärjestelmät
  - Windows
  - Mac OS X
- ❑ Komentoriviohjelmat
  - ping
  - tracert
- ❑ Ajurit
  - Tulostimen ajuri
  - Näytönohjaimen ajuri

# Konekieli ja ohjelmointikieli

---

- ❑ Kone ymmärtää vain "nollia ja ykkösiä"
- ❑ Ihmisen on työlästä kirjoittaa nollista ja ykkösistä koostuvaa ohjelmaa
- ❑ On kehitetty ohjelmointikieliä, joissa on omia kielioppeja
- ❑ Ohjelmointikielet voidaan kääntää tai tulkata konekielelle
- ❑ Ohjelmointikielillä voidaan siis määrätä kone tekemään jotain

# Erilaisia ohjelmointikieliä

---

- Java
- PHP
- Perl
- C
- C + +
- C#
- Delphi
- assembly
- Lisp
- Visual Basic
- Python
- Ruby
- JavaScript
- BASIC
- COBOL

# Java

---

- ❑ Java on Sun Microsystemsin johdolla 1990-luvulta alkaen kehitetty ohjelmointikieli
- ❑ Nykyisin Oraclen omistuksessa
- ❑ Java-kielellä voidaan tehdä ohjelmia työpöydälle, webbiin, mobiililaitteisiin, pesukoneisiin, pankkeihin,...
- ❑ Java nousi nopeasti suureen suosioon
- ❑ Työmarkkinoilla tarvitaan paljon Java-kieltä osaavia ohjelmoijia



# Ohjelman kirjoittaminen, kääntäminen ja tulkkaminen

---

- ❑ Lähdekoodi kirjoitetaan Javan kieliopin (eli syntaksin) mukaisesti tiedostoon Ohjelmani.java

```
public class Ohjelmani {  
    public static void main(String[] args) {  
        System.out.println("Heipparallaa");  
    }  
}
```

- ❑ Lähdekoodi käännetään tavukoodiksi komennolla  
**javac Ohjelmani.java**
- ❑ Hakemistoon muodostui uusi tiedosto  
**Ohjelmani.class**
- ❑ Tavukoodi ajetaan Java-tulkin avulla  
**java Ohjelmani**

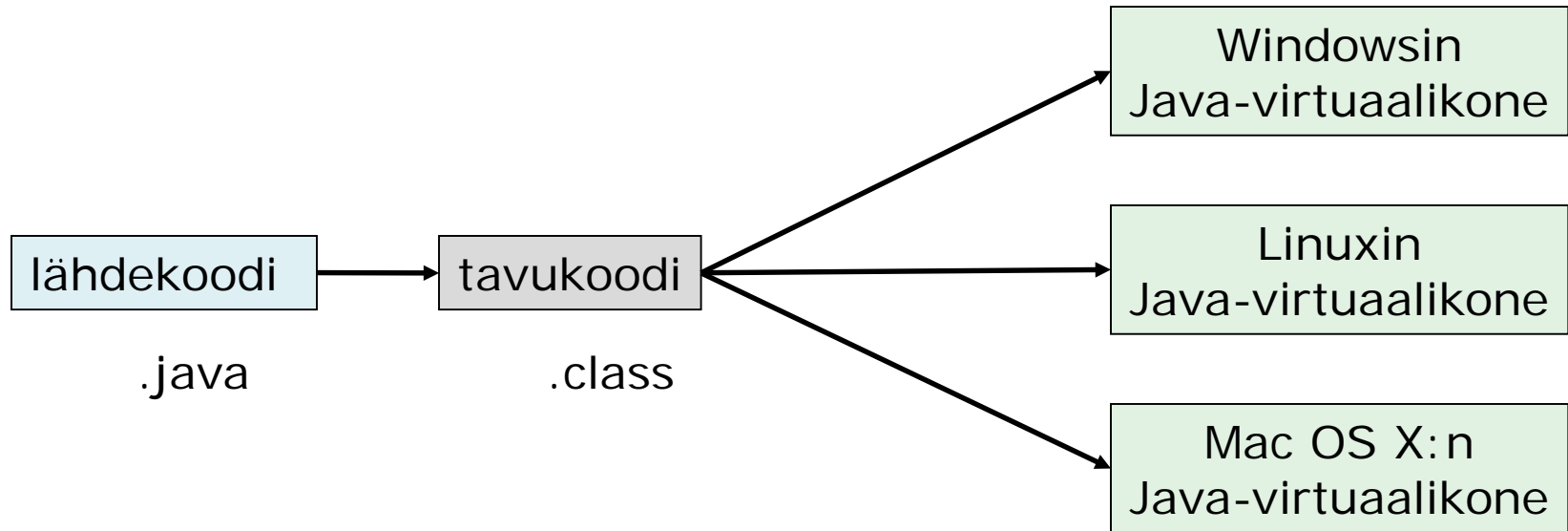
# Lähdekoodin ja tavukoodin kautta konekielelle

---

- ❑ Ohjelmani.java on lähdekoodia
- ❑ Ohjelmani.class on tavukoodia
- ❑ java-komento tulkkaa tavukoodin konekielelle ajonaikaisesti

# Java-ohjelman muodostaminen

---



# Skriptikielet ja käännettävät kielet

---

## □ Skriptikielet

- Esim. PHP
- Hidasta suorittaa
- Ei tarvitse kääntää erikseen
- Näppärää kehittää
- Tulkataan suoraan ohjelmointikielestä konekielelle ajettaessa

## □ Käännettävät kielet

- Esim. C++
- Nopeaa suorittaa
- Käännetään suoraan konekielelle
- Joudutaan kääntämään jokaiselle prosessorille/käyttöjärjestelmälle erikseen
- Ajetaan ilman tulkkia



# Tavukoodin tarkoitus

---

- Tavukoodilla ajettavat kielet
  - Esim. Java
  - Käännetään koneriippumattomalle tavukoodille
  - Tavukoodi muistuttaa lähemmin konekieltä kuin ohjelmointikieli ja on nopeampaa kuin skriptikielet
  - Jokaiselle konetyypille on olemassa oma tulkkinsa
  - Valmiista ohjelmasta riittää yksi julkaisu
- Java-tulkkia kutsutaan myös Javan virtuaalikoneeksi

# Johdanto Javaan

## Asennus

---

- Luokan koneille on asennettu JDK (Java SE Development Kit)
- JRE vrt. JDK
  - JRE (Java Runtime Environment)
    - Tarkoitettu kuluttajille
    - Sisältää pelkän Java-tulkin, jolla voi ajaa tavukoodiksi käännettyjä Java-sovelluksia
  - JDK (Java SE Development Kit)
    - Tarkoitettu ohjelmoijille
    - Sisältää Java-tulkin (JRE) sekä Java-kääntäjän (javac) ja muita apuvälineitä

# Johdanto Javaan

## Editorit

---

- Vaihtoehtoja
  - **Notepad, Notepad++**
  - **Eclipse**
  - **NetBeans**
  - TextPad
  - UltraEdit
  - ...
- Kehittyneiden editoreiden etuja
  - Syntaksin värikorostus
  - Kirjoituksen ennakoiminen
  - Koodin ulkoasun automaattinen muotoilu
  - Integroitu kääntäjä ja tulkkiin
  - ...

# Johdanto Javaan

## Java-kielen syntaksi

---

### EkaOhjelma.java


```
/**
 * Tämä on ensimmäinen ohjelmani
 */
public class EkaOhjelma {

    /**
     Ensimmäisen ohjelmani päämetodi
    */
    public static void main(String[] args) {

        // alla oleva lauseke tulostaa ruudulle
        System.out.println("Hello world!");

    }

}
```



# Java-kielen syntaksi

## Lähdekoodin kommentointi

---

- ❑ Ohjelmakoodin sekaan voidaan kirjoittaa muistiinpanoiksi ns. kommentteja
- ❑ Kommentit eivät vaikuta ohjelman suoritukseen millään tavalla
- ❑ Kommentit helpottavat lähdekoodin ymmärtämistä jälkikäteen
  - Kommentointi auttaa muita ihmisiä ymmärtämään kirjoittamaasi lähdekoodia
  - Kommentointi auttaa myös sinua ymmärtämään huomenna mitä ajattelit eilen
- ❑ Koodia kannattaa kommentoida lähes aina

# Java-kielen syntaksi

## Kommentit

Kommentin aloitusmerkintä

`/*` Tämä on kommentti, joka  
jatkuu seuraavilla  
riveillä. `*/`

Kommentin lopetusmerkintä

```
/**  
 * Tämäkin kommentti, joka jatkuu seuraavilla  
 * riveillä.  
 */
```

`//` Tämäkin on kommentti

```
// Kahdella kauttaviivalla aloitettu kommentti on voimassa  
// vain rivin loppuun asti
```

# Java-kielen syntaksi

## Whitespace

---

- Välilyöntien, rivinvaihtojen ja tabulaattorimerkkien peräkkäisellä määrällä ei ole toiminnallista merkitystä
- **Koodin sisennystä ja rivinvaihtoja käytetään lähdekoodin luettavuuden parantamiseksi.**
- Sanoja ei voi kuitenkaan kirjoittaa yhteen

# Java-kielen syntaksi

## Luokka

---

- Java-ohjelma kirjoitetaan aina johonkin luokkaan.
- **Luokan nimi aloitetaan isolla kirjaimella.**
- Esimerkin tapauksessa luokan nimi on EkaOhjelma ja se on määritelty julkiseksi.

```
public class EkaOhjelma {    ...  
  
}
```

- Luokka määritellään varatulla sanalla class ja luokan julkisuus sanalla public.
- Näiden jälkeen tulevat luokan nimi ja lohkon aloittava aaltosulku {.
- Tämän jälkeen tulisivat kaikki luokkaan kuuluvat metodit ja muuttujat.
- Lohkon päättävä aaltosulku } päättää luokan määrittelyn.



# Java-kielen syntaksi

## Päämetodi

---

- Esimerkissä luokkaan on kirjoitettu vain yksi metodi eli päämetodi.

```
public static void main(String[] args) {  
    ...  
}
```

- Sinun ei tarvitse vielä ymmärtää sen tarkoitusta.
- Ensimmäisten ohjelmiesi koodi kirjoitetaan päämetodin aaltosulkujen sisään
- Kirjoita päämetodin otsikko toistaiseksi aina näin

# Java-kielen syntaksi

## Tekstin tulostaminen konsoliin

---

- ❑ Toiminnallisuus kirjoitetaan siis päämetodin aaltosulkujen väliin
- ❑ Esimerkkiohjelman toiminnallisuus on saatu mahtumaan yhdelle riville.

```
System.out.println("Hello World");
```

- ❑ Sulkujen sisään kirjoitetaan lainausmerkeissä tulostettava merkkijono
- ❑ Rivin alun tarkoitus (System.out.println) selviää myöhemmin
- ❑ Tärkeää on ymmärtää, että kyseinen käsky tulostaa näytölle tekstiä
- ❑ Huom! Javassa lauseet päätetään ';' -merkkiin eli puolipisteeseen, koska rivinvaihdolla ei ole merkitystä.