

## **R0027 – Ohjelmoinnin perusteet Kehittämisharjoitukset**

### **A: Johdatus ohjelmointiin ja Java-ohjelmointikieleen**

*Luokan koneille on asennettu valmiiksi JDK, ECLIPSE ja muut tarvittavat ohjelmat.*

#### **JAVA YMPÄRISTÖ (KOMMENTORIVI JA NOTEPAD KEHITYSYMPÄRISTÖNÄ)**

##### **1. Check that JDK is installed in your computer**

Using Windows File Explorer check that the JDK files are installed in your computer. If the installation was done to the default directory, you will find the JDK at C:\Program Files\Java

##### **2. Check that you can run JDK's tools from your computer**

Open Windows' command prompt (cmd.exe), write *javac* and press *<enter>*.

If the JDK is installed and configured properly you should see the following output after running javac:

```
ca. C:\windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\ANTDEAR>javac
Usage: javac <options> <source files>
where possible options include:
    -g                  Generate all debugging info
    -g:none             Generate no debugging info
    -g:<lines,vars,source> Generate only some debugging info
    -nowarn             Generate no warnings
    -verbose            Output messages about what the compiler is doing
    -deprecation        Output source locations where deprecated APIs are used
    -classpath <path>   Specify where to find user class files and annotations processors
    -cp <path>          Specify where to find user class files and annotations processors
    -sourcepath <path>  Specify where to find input source files
    -bootclasspath <path> Override location of bootstrap class files
    -extdirs <dirs>     Override location of installed extensions
    -endorseddirs <dirs> Override location of endorsed standards path
    -proc:<none,only>   Control whether annotation processing and/or compilation is done.
    -processor <class1>[,<class2>,<class3>...] Names of the annotation processors to run; bypasses default discovery process
    -processorpath <path> Specify where to find annotation processors
    -parameters         Generate metadata for reflection on method parameters
    -d <directory>      Specify where to place generated class files
    -s <directory>      Specify where to place generated source files
    -h <directory>      Specify where to place generated native header files
    -implicit:<none,class> Specify whether or not to generate class files for implicitly referenced files
    -encoding <encoding> Specify character encoding used by source files
    -source <release>    Provide source compatibility with specified release
    -target <release>    Generate class files for specific VM version
    -profile <profile>   Check that API used is available in the specified profile
    -version            Version information
    -help              Print a synopsis of standard options
    -Akey[=value]       Options to pass to annotation processors
    -X                 Print a synopsis of nonstandard options
    -J<flag>            Pass <flag> directly to the runtime system
    -Werror             Terminate compilation if warnings occur
    @<filename>         Read options and filenames from file

C:\Users\ANTDEAR>_
```

### 3. Muuta Windowsin asetukset näyttämään tiedostopäätteet

Control Panel -> Appearance and personalization -> Folder Options -> View ->  
Merkitse "*Hide extensions for known file types*" pois päältä.

### 4. Hae Optimasta tiedosto Hei.java, tutustu lähdekoodiin, käännä se tavukoodiksi ja aja tavukoodi Java-tulkilla

Luo H-asemalle uusi kansio, jonka nimi on "*java\_test*"

Hae Optimasta tarvittava tiedosto Hei.java.

Tallenna tiedosto verkkolevylle hakemistoon H:\java\_test\

Avaa komentorivi ja mene kyseiseen kansioon:

h: <enter>

cd java\_test <enter>

Avaa lähdekoodi tarkasteltavaksi notepad-ohjelmalla:

***notepad Hei.java <enter>***

Tutustu lähdekoodiin.

Sulje Notepad

Käännä lähdekoodi tavukoodiksi:

***javac Hei.java***

Check that a file named Hei.class has been created in your H:\java\_test\ directory.

Aja käännetty tavukoodi (Hei.class) java-tulkin avulla:

***java Hei***

## 5. Laadi toinen ohjelma, joka tervehtii sinua omalla nimelläsi.

Kopioi Hei-ohjelman lähdekoodi pohjaksi uudelle ohjelmalle.

Kirjoita uuden ohjelman lähdekoodi tiedostoon Hei2.java

Muuta ohjelman nimi myös lähdekoodiin.

Muokkaa siis rivi

public class Hei {

muotoon

public class ***Hei2*** {

Muokkaa merkkijonon "Tervehdys Jaakko!" tilalle oma tervehdyksesi.

Käännä ja aja uusi ohjelmasi.

Komentoi ohjelman koodia siten, että lisäät ohjelman alkuun usean rivin mittaisen kommentin, jossa kerrot nimesi, opiskelijanumerosi sekä päivämäärän. Usean rivin kommentti aloitetaan /\* - merkinnällä ja lopetetaan \*/ -merkinnällä.

Lisää jokaisen ohjelmarivin yläpuolelle yhden rivin kommentti, jossa kirjoitat itsellesi (tai jollekin muulle kehittäjälle) kuvauksen siitä, mitä seuraava rivi tekee. Yhden rivin kommentti aloitetaan // merkinnällä. Sitä ei tarvitse lopettaa, sillä se on voimassa koko loppurivin ajan.

Käännä ja aja uusi ohjelmasi.

## 6. Hae Optimasta tiedosto GuiHei.java. Tutustu ohjelmaan, käännä ja aja se.

## 7. Tee uusi sovellus, joka avaa ikkunan, jonka sisältönä on lempisananlaskusi.

# ECLIPSE KEHITYSYMPÄRISTÖNÄ

**Luo työpöydällesi pikakuvake, joka osoittaa Eclipse-ohjelmaan**

[hiiren oikealla napilla] työpöydän tausta -> new -> shortcut -> C:\Eclipse\eclipse.exe

## 8. Käynnistä Eclipse-ohjelma luomaasi pikakuvaketta käyttäen. Työtilaksi sopii esim.

**"H:\eclipse\_workspace"**

Eclipse käyttää tätä hakemistoa nyt projektien ja ohjelmien tallennukseen.

9. **Sulje Welcome-välilehti**

10. **Tee Eclipsellä uusi Java-projekti nimeltään ”Kehittamisharjoitus\_A”.**

File -> New -> Java Project  
Project name = ”*Kehittamisharjoitus\_A*”  
Finish.

11. **Luo em. projektiin uusi luokka Eka (java-tiedosto ”Eka.java”), joka tulostaa komentoriville sanan eka.**

Aktivoi em. projekti hiiren vasemmalla painikkeella.

File -> New -> Class  
Name = *Eka*  
(Which method stubs would you like to create?) public static void main(String[] args) = *Päälle*  
Finish.

Kirjoita rivin “// **TODO** Auto-generated method stub” tilalle:  
*System.out.println("eka");*

12. **Aja juuri luomasi ohjelma Eclipsen avulla**

kääntäminen tapahtuu automaattisesti, kun  
Project->Build automatically  
on asetettu päälle.

Klikkaa hiiren oikealla napilla Eclipsessä luomaasi ohjelmaa Eka -> Run as... -> Java Application

13. **Varmista, että Eclipsen sisäiseen konsoliin tulostuu oikea teksti**

Eclipsen Java-perspektiivissä on alhaalla Console-välilehti, jossa komentorivitulosteet näkyvät.