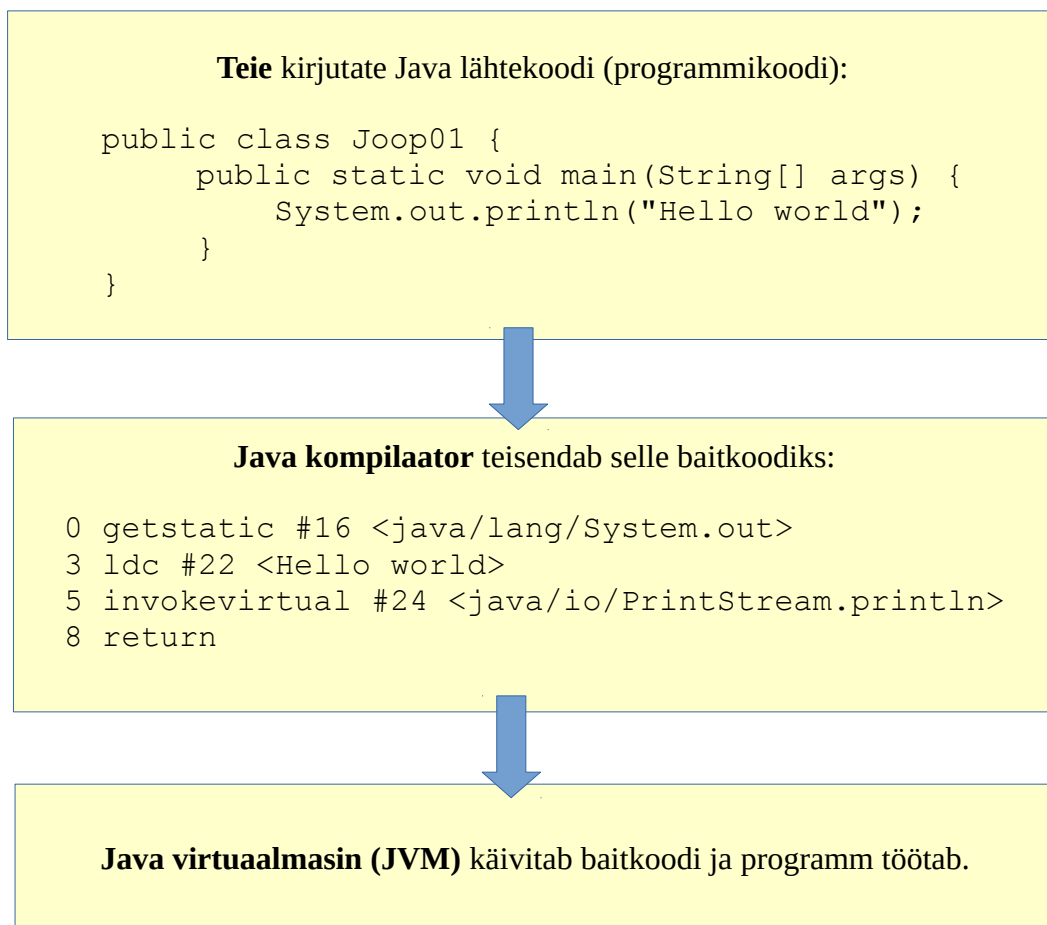


Kuidas Java töötab? Mis on OOP?

Tänase praktikumi eesmärgiks on uurida Java töö põhimõtet ning teha vahet objekti meetodil ja staatilisel meetodil ning väljadel.

Teooria

Programmikoodi ei saa otse operatsioonisüsteemis käivitada. Järgimine joonis näitab, kuidas Java programmikoodist käivitamiseni jõuda.



Java baitkoodi vaatamiseks on mitmeid võimalusi, üks mugav võimalus on kasutada jclasslibi: <https://github.com/ingoegel/jclasslib>

Ülesanne 1

Kirjutage tavalise tekstiredaktoriga (nt Notepad++) lihtne Java programm ning kompileerige ja käivitage see käsurealt. NB! Notepad++ puhul valige failide encodinguks Encoding menüüst „UTF-8 without BOM”. BOM märgiga kood ei pruugi kompileeruda.

Teil tuleb kirjutada lihtne Java programm *main* meetodiga, kus trükite konsooli: „See on staatiline

meetod, see ei ole objekt. Uhtlasi on see main meetod.” Vihje programmi kirjutamiseks on üleval diagrammil (vahetage tekst ära).

NB! Failinimi pange sama, mis klassinimi. See pole küll tingimata nõutud, aga hea tava, mida siin kursuses järgime. Klassinimi kirjutage suure tähega (failinimi ka)!

Java programmi kompileerimiseks (programmikoodi teisendamiseks baitkoodiks) tuleb kasutada Java kompilaatorit. Teil on palju mugavam kui Java kompilaator on PATHis (kooli arvutis te seda ise seadistada ei saa):

Esiteks tehke kindlaks, kus asub Java kompilaator javac. Tavaliselt asub ta JDK kataloogis, näiteks: C:\Java\jdk1.8.0_25\bin – see kataloog tulebki lisada PATHi. NB! Teie arvutis võib JDK (Java Developers Kit) olla installeeritud teise kohta, nt Program Files (x64) alla – otsige üles!

JDK allalaadimine: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Enda arvutis tehke kindlaks, kas JDK bin kataloog on pathis. Avage käsurida (cmd) ja kirjutage PATH. Kui seda ei ole, siis:

Lisage JDK bin kataloog pathi, õpetus:

<http://www.howtogeek.com/118594/how-to-edit-your-system-path-for-easy-command-line-access/>

1. Avage käsurida (cmd) navigeerige „cd” käsuga kataloogi, kus teie loodud programmikood asub.
2. kompileerige Java programmikood käsuga: *javac filename.java*
3. käivitage kompileeritud klassifail javaga (mitte kompilaatoriga!): *java klassinimi* (ilma laiendita)

Tulemuseks peab programm välja trükkima enne mainitud teksti.

Taust

Loome lihtsa tudengite infosüsteemi, mis salvestab iga tudengi kohta läbitud ainepunktide arvu. Teeme seda kahel viisil – esmalt kasutade staatilisi meetodeid (protseduurilist programmeerimist) ja seejärel objektorienteeritult.

Kui te kirjutate Javas klassi, siis üldjuhul selleks, et luua klassist objekte. Kuid on erijuhte – näiteks ainult staatilisi meetodeid sisaldavad klassid. Neid saate kasutada ilma objekti loomata. Staatilised meetodid on seotud klassiga, mitte klassist loodud objektiga. Staatilist meetodit (ehk klassimeetodit) iseloomustab *static* võtmesõna meetodi signatuuri ees.

Ülesanne 2

2.1 Kõigepealt protseduuriline versioon: looge klass StudentProc, kus on meetodid:

```
public static void addStudent (String name) – uue tudengi lisamiseks süsteemi  
public static void addPoints(String studentName, int pointsToAdd) –  
tudengile ainepunktide lisamiseks  
public static int getPoints(String studentName) – ainepunktide arvu  
küsimiseks
```

2.2 Täiendage esimeses ülesandes loodud programmi: looge seal 3 uut tudengit, määrake neile

ainepunkte ja trükkige `getPoints()` meetodi abil ainepunktid konsooli.

2.3 Kompileerige ja käivitage programmikood. Käivitama peate endiselt ainult ülesandes 1 loodud klassi, sest seal on ainsana main meetod. Kompileerida on vaja kõik klassid.

NB! Korraga saate kompileerida ka mitu klassi:

```
javac Klass1.java Klass2.java
```

Üleanne 3

Eelnevalt nägite, et staatilist meetodit saab välja kutsuda ilma objekti loomata, just nagu tavalist funktsiooni mitte-objektorienteeritud keeltes. Nüüd aga vaatame, mis teeb Javast objektorienteeritud keele – looge sama funktsionaalsus objektorienteeritult.

3.1 Looge uus klass `StudentOOP` ja sinna konstruktor ning kaks meetodit:

```
public void addPoints(int pointsToAdd) – tudengile ainepunktide lisamiseks  
public int getPoints() – ainepunktide arvu küsimiseks
```

3.2 Täiendage esimeses ülesandes loodud programmi: looge seal 3 uut tudengit, määrake neile ainepunkte ja trükkige `getPoints()` meetodi abil ainepunktid konsooli.

3.3 Kompileerige ja käivitage programmikood. Käivitama peate endiselt ainult ülesandes 1 loodud klassi, sest seal on ainsana main meetod. Kompileerida on vaja kõik muudetud klassid.

Ülesanne 4

Looge esmalt objektorienteeritud versioonis ja seejärel staatilises versioonis juurde sama nimega tudeng, kes teil juba olemas on. Nt kui teil on olemas tudeng Mary, siis looge teine Mary veel. Kumbas versioonis on lihtsam tööle saada, et olemas oleks 2 Maryt korraga?