



1918

**TALLINNA TEHNIKAÜLIKOOL**  
TALLINN UNIVERSITY OF TECHNOLOGY

# **IDK0051 Objektorienteeritud programmeerimine Javas**

Martin Rebane (martin.rebane@ttu.ee)

# Täna

- Mida me aines õpime?
- Aine läbimise tingimused
- Objektorienteeritud programmeerimine
- Objektid
- Nähtavus
- Java virtuaalmasin (JVM)



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Aine

- Objektorienteeritud programmeerimine Javas
  - kuni 2017/2018 õa
- Infosüsteemide arendamine Javas
  - Alates 2018/2019 õa



1918

**TALLINNA TEHNIKAÜLIKOOL**  
TALLINN UNIVERSITY OF TECHNOLOGY

# Miks OOP?

- Arendame tarkvara, mis vastaks kasutaja või tellija ootustele
- Kasutajate ootused muutuvad ja süsteemid kasvavad aja jooksul
- Objektorienteeritud arhitektuur võimaldab muutusi lihtsamini läbi viia ja hallata



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Praktiline info

- Aine koduleht IT teaduskonna õppeainete lehel (IDK0051): <https://ained.ttu.ee/>
- Registreerumise võti praktikumis
- Deklareerige praktikumi õppejõu nimele



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Aine korraldus

- 8. nädalal (23.-29. oktoober) loengut ja praktikume ei toimu
- Võimalik, et mõned loengud toimuvad videoloengutena (aga võimalik, et ka mitte)



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Aine korraldus

- Loengutestid
  - 7. nädala loengu lõpus
  - 16. nädalal loengu lõpus
  - Järgi teha ei saa (vt mõjuval põhjusel **etteteatamisega**), läbikukkumisel aine lõpus teooriaeksam
- Kontrolltööd
  - 7. nädalal
  - 15. nädalal



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Aine korraldus

- Eksam – iseseisev töö kaitsmisega
- Nõuded aine läbimiseks:
  - 51% teooriapunkte
  - 51% mõlemast kontrolltööst eraldi
  - 51% eksamitööst
- Täpsed reeglid [ained.ttu.ee](http://ained.ttu.ee) keskkonnas



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY



# Põhiteemad

1. Objektid ja disain
2. Realisatsiooni varjamine, API disain
3. Andmetüüpide disain, vood
4. Polümorfism, ülelaadimine, ülekirjutamine
5. Java Virtuaalmasina (JVM) tööpõhimõtted
6. Liidesed, erindid. Tüübituvastus.
7. Kapseldamine. Kompositsioon.
8. Töö lõimesed (*threads*), paralleeltöö
9. Predikaadid, lambdad. Spring Boot.



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Läbimise nõuded ja hinde kujunemine

- 1 p taustatest 4. nädalal
- 20 p kodutööd (9 tk)
- 40 p kontrolltööd (20 p x 2 tk)
- 20 p teooriatestid (10 p x 2 tk)
- 
- 20 p eksamitöö ja selle kaitsmine
- Boonuspunktid suurepärase soorituse eest



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Kuidas seda ainet õppida?

- Osalege võimalikult palju loengutes ja praktikumides
- Planeerige igal nädalal aeg kodutööks
- Kui soovite nõu küsida või hakkate maha jääma, tulge kohe konsultatsiooni!
- Küsige kolleegidelt nõu, kuid ärge tehke copy-pastet



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Kui teil on huvi kohtuda õppeprodekaaniga...

- Kui tahate temaga ühendust saada, siis palju lihtsam on, kui kirjutate talle otse [Margus.Kruus@ttu.ee](mailto:Margus.Kruus@ttu.ee)
  - Tülikam võimalus – spikerdamise/plagiaadiga vahele jäädes aitan teil ise kohtumise korraldada. Hea meelega väldiksin seda vahemehe rolli :)
  - Vt tagajärgi ka TTÜ õppekorralduse eeskirjast



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

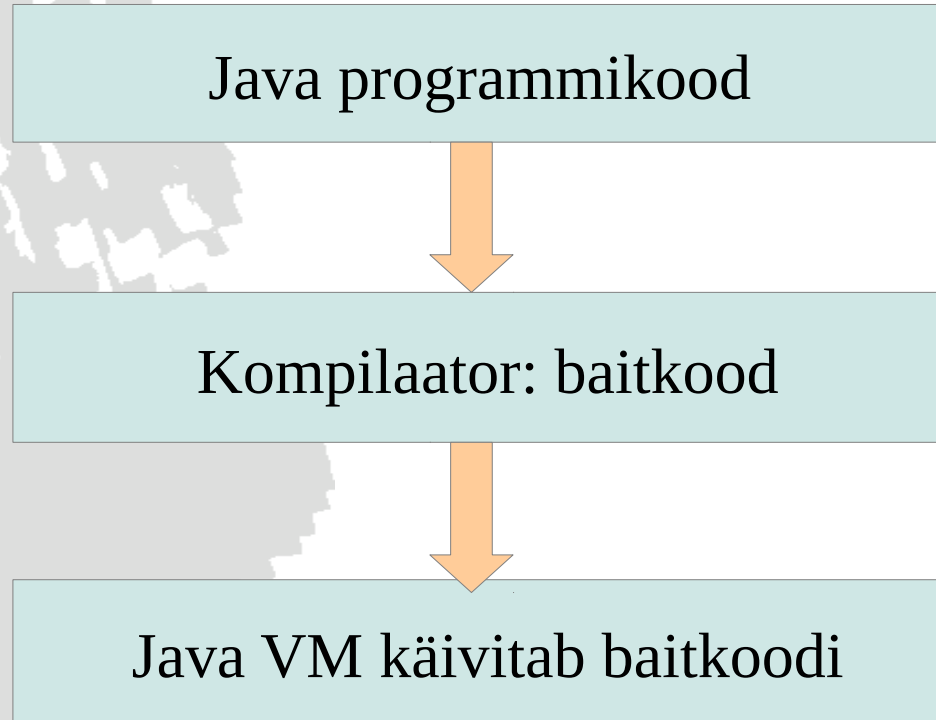
# JAVA ja JVM



1918

**TALLINNA TEHNIKAÜLIKOOL**  
TALLINN UNIVERSITY OF TECHNOLOGY

# Koodist töötava programmini



kirjutate Javas  
Python,  
Coldfusion

javac  
Eclipse compiler  
in JDT Core

Java Hotspot  
Azul, JRockit jt

# Objektorienteeritud programm

- OO programm on objektide kogum, mis suhtlevad omavahel sõnumeid saates ja vastu võttes (meetodi väljakutsega)
- Soovime peita selle, kuidas mingi eesmärk saavutatakse



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Kordamine - põhimõisted

- Klass = tüüp, definitsioon (programmikood). Olemas ka siis kui programm parasjagu ei tööta.
- Objekt = klassist tehtud konkreetne eksemplar programmi töö ajal (new Type..)



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY



# Objekti loomine

- Mille tekitab järgmine programmikood: `Student s = new Student(17);`

- Mis on `s`?
- Mis on `Student`?
- Mis on `Student()`?
- Mis on `17`?

Viide objektile

Loodava objekti tüüp

Konstruktori väljakutse, tagastab viite objektile

Argument konstruktorile



# Static typing

- Muutuja tüüp on teada kompileerimise ajal – seda muuta ei saa
  - Vähendab vigade hulka programmi töö ajal
  - Java, C++
- Alternatiiv: dynamic typing
  - Muutuja tüüp selgub programmi tööajal (runtime interpretation)
  - Python, PHP



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Mis on argumendi tüüp?

- Tüpiseerimata keeles pole vahet, kas salvestame muutuja külge numbri, teksti või massiivi:

```
<?php  
    $var = 5;  
    $var = "Hi!";  
    $var = array();  
?>
```



# Duck typing

- Objektilt eeldatakse mingite omaduste olemasolu, nt meetodi `run()`
- Objekt ei pea olema mingit kindlat tüüpi, kui omadus on olemas, nt meetod `run()`. See käivitatakse.
- Vajadus Duck-typingu järgi peaks tekkima üsna harva. Javas saab kasutada reflectioni abil



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Java ja typing

- Static typing – staatiliselt tüpiseeritud - jah
  - `int a = 5;`
  - ~~`a = "Tere";`~~
- Dynamic typing – dünaamiliselt tüpiseeritud – ei
  - `int a = 5;`
  - `a = "Tere";`
- Duck typing – jah, *reflectioni* abil, aga pole hea tava



# Klass

- Konstruktor (ei pruugi olla, siis loob kompilaator selle teie eest)
- Väljad (muutujad)
- Meetodid (funktsioonid)



1918

**TALLINNA TEHNIKAÜLIKOOL**  
TALLINN UNIVERSITY OF TECHNOLOGY

# Nähtavus

- Iga klass, väli ja meetod omab nähtavust
- Ülemise taseme klassil:
  - public – nähtav kõigile
  - package-private – nähtav paketi sees
- Alumise taseme elemendid (meetodid, väljad, sisemised klassid) lisandub:
  - private – nähtav klassi sees
  - protected – nähtav paketi sees ja alamklassidele



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Nimetamine

- Konstruktoril on sama nimi, mis klassil!

- NB! Ka failinimi sama, mis klassil:

```
MyClass.java  
class MyClass  
public MyClass() {
```

sUur- ja vÄiKeTäheD  
oN OlulIseD



# Koodi vormindus

- Kuidas vormindada Java koodi?
- Klassi nimi suure tähega CamelCase
- Konstruktori nimi samamoodi
- Failinimi sama, mis klassil
- Meetodid väikese tähega camelCase
- Igasse faili kirjutage üks klass

NB! Seda vormindust  
eeldame iga esitatud  
töö juures!



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Nähtavus

## Access Levels

Modifier	Class	Package	Subclass	World
<code>public</code>	Y	Y	Y	Y
<code>protected</code>	Y	Y	Y	N
<i>no modifier</i>	Y	Y	N	N
<code>private</code>	Y	N	N	N

<http://docs.oracle.com/javase/tutorial/java/javaOO/accesscontrol.html>

# Nähtavus JDK 9

- Lisandub moodule (*module*)
  - Sh liigendatakse ka JDK enda moodulid



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Konstruktor

- Lihtne seletus:
  - Sama nimega, mis klass
  - Tagastatava väärtuse tüüpi ei määrata
  - Käivitatakse alati kui klassist objekt luuakse
- Lisainfo:
  - Tagastab **viite loodud objektile**
  - Kutsus alati välja ülemklassi konstruktorit

Mida konstruktor tagastab?



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Staatiline meetod ja väli (static)

- Seotud klassiga, mitte objektiga
- Käitumine igal objektil ühesugune
- Saab kasutada ilma objekti loomata
- Staatilise meetodi sees ei saa kasutada objekti välju ega meetodeid:  
Cannot make a static reference to the non-static field X



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Static

- Staatileine meetod: kasutage nt algoritmimisel, valemite realiseerimisel jmt
- Staatileine väli: üle süsteemi ühtne väärtus, võib kasutada ilma objekte loomata või objektide kontekstis
- Kasutamine ilma objektita:
  - `Klass.staticMethod();`  
`Klass.staticField`



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Suurem pilt ja Java



1918

**TALLINNA TEHNIKAÜLIKOOL**  
TALLINN UNIVERSITY OF TECHNOLOGY

# Kus Java asub?

- Protseduurilised keeled
  - Jadas täitmine, protseduurid (funktsioonid), nt Basic
- Objektorienteeritud keeled
  - Suhtlus objektide vahel
  - Objekt kui andmetüüp
- Funktsionaalne programmeerimine
  - Clojure, F#, Java 8
  - Vältib olekuid

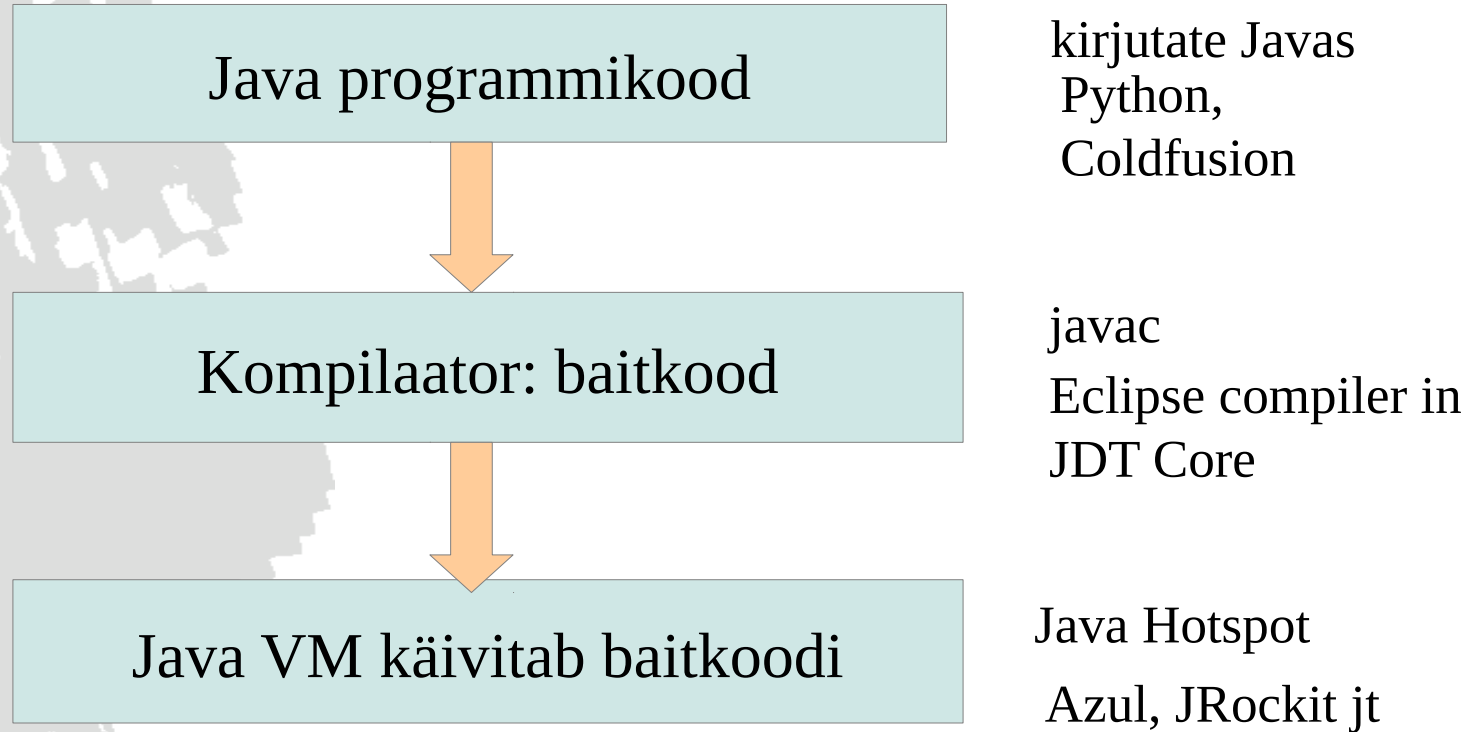


1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY



# Koodist töötava programmi



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Terminid

Java programmi **käivitamiseks**

- JRE – Java Runtime Environment, sh virtuaalmasin Java käivitamiseks (java), teegid jms

Java programmi **arendamiseks**

- JDK – Java Development Kit, arendusvahend (kompilaator javac, tööriistad, sisaldab ka JRE asju)



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Kui palju tööd Javas teha saab?

- Piiratud masina võimalustega:
  - Mälumaht
  - Protsessori jõudlus
  - I/O kiirus (kõvaketas, võrk vmt)
- Arendaja võimalused:
  - Koodi optimeerimine
  - Paralleeltöö lõimede abil



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Mis mõjutab programmi jõudlust?

- Loodud ja mälus hoitavate objektide arv ja suurus.  
Kuidas mõjutada?
  - Voogude kasutamine andmete lugemisel
  - Optimaalsete andmestruktuuride valimine (millist kollektsiooni, massiivi vm kasutada)
  - Programmi arhitektuur (kuidas objektid omavahel suhtlevad)
  - Andmete kopeerimine vs viitamine

# Java programmeerimiskeel

- Kõik asjad on objektid
  - OOP juhtmõte
  - aga kas on vaja kõik objektiks teha (mis juhtub jõudlusega?)
- Kõike ei ole vaja objekti kaudu kasutada:
  - Staatilised meetodid/väljad
  - Lambda-funktsioonid



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# OOP paradoks

- Kõik asjad on objektid, aga hoame objektide hulga madala (Bloch)
  - *static factory method* konstruktori asemel – võimaldab kontrollida konstruktori kasutamist, sama objekti taaskasutada või objekti mitte luua
  - *Singleton property* – loome ainult ühe objekti ja taaskasutame seda
  - Kui klassis on ainult staatilised meetodid, siis keelame objekti loomise ära privaatsel konstruktoriga



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Objektide suhtlus

- Objektorienteeritud programm – omavahel suhtlevad objektid
- Objektide käitumiskooodeksid?
- Disainimustrid – standardised viisid objektide omavaheliseks suhtluseks



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Objekt ei ole alati vajalik!

- Enamasti on objektide kasutamine väga selge ja lihtne, kuid mõnel juhul täiesti mittevajalik (nt anonüümsed klassid)
- Javas on ka muud *boilerplate* koodi
- Java 8 pakub lahenduse: funktsionaalse programmeerimise elemendid



# Aine orienteeruv ülesehitus

- 1.-2. nädal: kordamine ja baas
- 3.-4. nädal: polümorfism
- 5.-7. nädal: OOP printsiibid
- 9.-11. nädal: vood, funktsionaalne prog.
- 12.-16. nädal: paralleelne prog.
- ... ja muud teemad nende vahel.



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# IDE

- Kasutage millist soovite, minu näited Eclipsega
- Eelmiste aastate kogemuse põhjal
  - IDEA fännidele: tõmmake Eclipse installatsioon varuks kui IntelliJ crashib, praktsi ajal pole aega seda remontida :)
  - Päriselt ka!



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# JDK

- Installeerige JDK 8
- JDK 9 *release date* on 21. september
  - Kindlasti läheme üle selle semestri jooksul



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Pange java ja javac pathi!

- ... või õppige ilma kasutama.
- Google: add java to path Windows/Linux/Mac
- NB! Lisaks javale kasutame ka **javac**



1918

TALLINNA TEHNIKAÜLIKOOL  
TALLINN UNIVERSITY OF TECHNOLOGY

# Esimeseks praktikumiks

- **Installeerige JDK ja IDE**
- **Vaadake, et java ja javac oleks PATHis!**
- Küsimused: [martin.rebane@ttu.ee](mailto:martin.rebane@ttu.ee)
- Deklareerige praktikumi õppejõu nimele?? Selgub.



1918

**TALLINNA TEHNIKAÜLIKOOL**  
TALLINN UNIVERSITY OF TECHNOLOGY