

Java Fundamentals

Lecture 1 - Introduction

Sven Laanela
[@svenlaanela](#)

30.01.2017

A **long** time ago, in a galaxy **far** away

- **2012** - Tartu University - **Java Fundamentals** - Java alusehitus
- **2012, 2014, 2016** - Tartu University
- **2013** - Tallinn Technical University

Fundamentals?

Fundamentals?

- Not a beginners course!
- Course for students **who think they already know Java**
- We'll go **deep into Java** to understand **how it actually works**
- Become a **better programmer**

Why are **we doing this?**

ZeroTurnaround

ZeroTurnaround

- Started in **2007**
- Builds **tools** for **Java Developers** - **JRebel, XRebel**
- Need to really understand **how Java/JVM works**
- Engineers tackle **hardest issues on the platform...**
- ... and **discuss these over lunch** :)

Why are **we** doing this?

- We love Java and like to **spread this love!**
- Java is a **great technology** to invest in!
- Our small way of **giving back** to the community!



**Why are YOU doing
this?**

Team

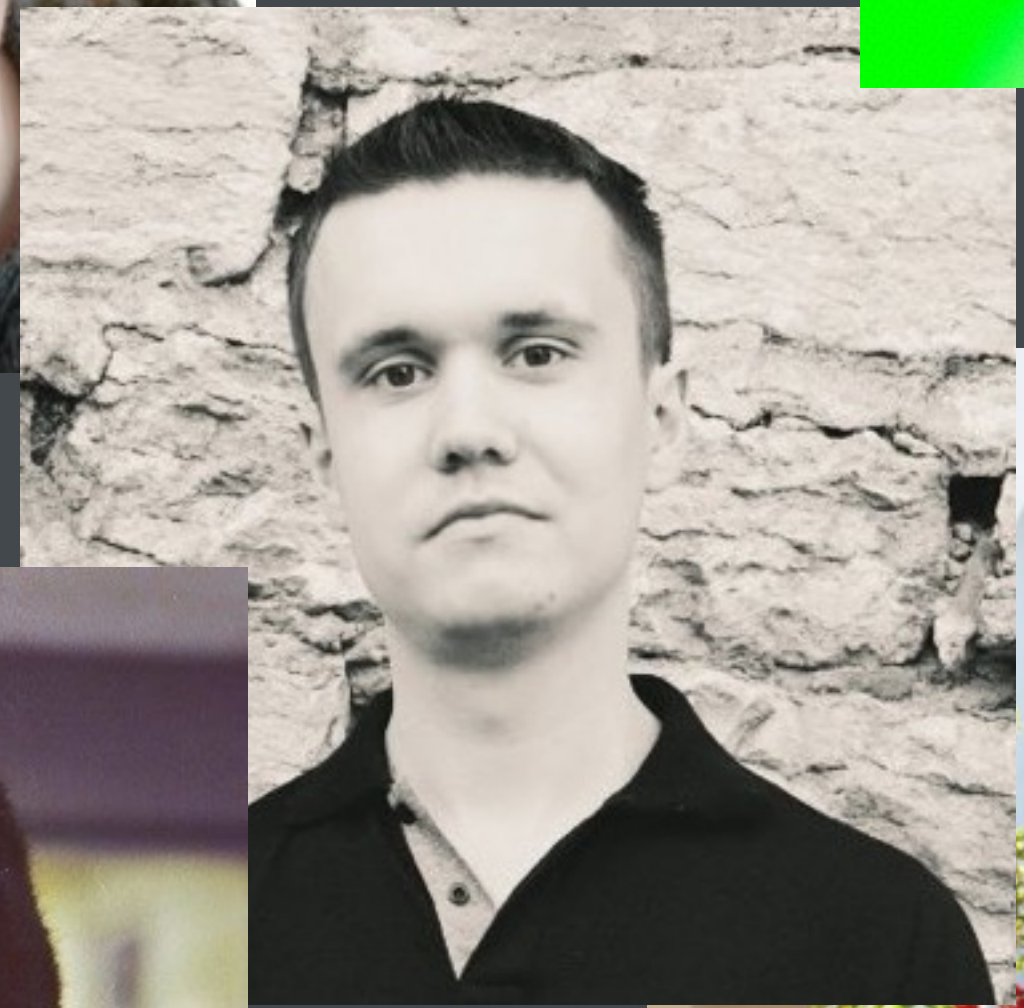
Anton



Bram



Mihhail



Oleg



Tõnis



Simon



Arnel



Tom



Sven



Course

- **ITI8905**
- **3 EAP** ~ 78 hours of work
- **15 lectures**
- **14 homework** assignments (*lectures 1-14*)
- **Mondays @ 12:00-13:30** @ ICT-315 (*Akadeemia tee 15a*)
- **Info** @ <https://courses.cs.ttu.ee/pages/ITI8905>
- **Contact** jf@zeroturnaround.com

Expectations

- You understand **Java**
- Know how to use an **IDE** - *Eclipse, IntelliJ IDEA, NetBeans*
- Know about **Maven**
- Can read, write and understand **English** :)
- Know how to **send emails**

Expectations

- **JDK8**
- **Maven (3.3)**
- **IDE**
- **JUnit**
- **Git**

Warning!

- **40** seats, register early
- This course is **tough**, so only register if you do want to take it!
- And don't forget to deregister early if you don't! ;)
- Don't be here for points
- Last time - **45** registered, **24** passed

Course outline

- **Lecture 1** - *Jan 30* - **Course Introduction**
- **Lecture 2** - *Feb 6* - **Lambdas & Streams**
- **Lecture 3** - *Feb 13* - **Collections & Generics**
- **Lecture 4** - *Feb 20* - **Java I/O**
- **Lecture 5** - *Feb 27* - **Applications of Java I/O**
- **Lecture 6** - *Mar 6* - **Threads & Java Memory Model**
- **Lecture 7** - *Mar 13* - **Threads: Thread Safety & Locks**
- **Lecture 8** - *Mar 20* - **Concurrency API**

Course outline

- **Lecture 9** - *Mar 27* - **Java Networking**
- **Lecture 10** - *Apr 3* - **Java Memory Management & Garbage Collection**
- **Lecture 11** - *Apr 10* - **ClassLoaders**
- **Lecture 12** - *Apr 17* - **Reflection API and Dynamic Proxies**
- **Lecture 13** - *Apr 24* - **Java bytecode, Javassist**
- **Lecture 14** - *May 8* - **Java Troubleshooting, Performance**
- **Lecture 15** - *May 15* - **Java 9 and future**

Final grade

- **50% Homework**
- **50% Exam**
 - **60%** written (multiple-choice)
 - **40%** oral (3 questions)
 - **Prerequisite**: must get at least **6 points** for Homework

Homework

- **14 assignments total**
- Each assignment scored **0.0 - 1.0**
- **Max** total score is **12**
- **Exam prerequisite:** get at least **6 points** for homework!
- **Bonus:** get **more than 12 points** - have an easier time on the exam

Homework

- Given out at the end of the lecture (Monday)
- **Hard deadline** is **Sunday 23:59 EEST** same week **(6 days)**
 - **If you are late** then you'll get **0 points!**
 - **If you are late** then we won't even look at it!
- Deadline for this lecture's is **Sunday, February 12 23:59 EEST**

Homework

- **No cheating!**
- **Copying solutions from a friend** is forbidden - if we see two identical submissions, they **both get a 0.**
- **Copying solutions from StackOverflow** is forbidden - if we detect this we will deduct points.
- Use the **Java Standard Library** unless homework assignment itself uses some other libraries

Homework demo

Homework #1

- Link: <https://github.com/JavaFundamentalsZT/jf-hw-intro>
- Goals:
 - Get the failing unit test to pass
 - Submit it
- Deadline: **Sunday, February 12 23:59 EEST**

Homework #1

- `git clone https://github.com/JavaFundamentalsZT/jf-hw-intro`
- `cd jf-hw-intro`
- `./mvnw clean package` - *this should fail initially and succeed after fixing test*
- `java -jar target/jf-homework1.jar firstString secondString` - *optional*

Homework #1

- `./mvnw clean deploy`

Your full name (e.g. John Smith):

Jane Smith

Your Student Code (matrikuli number, e.g. ABCD12345):

ABCD012345

Comment:

Java IO

- Attach **target/jf-homework1-ABCD012345.zip** and submit to jf@zeroturnaround.com

Puzzlers :))

Guess the output #1

```
System.out.println(10/3);
```

```
System.out.println(10/6);
```

Guess the output #1

```
System.out.println(10/3); // == 3
```

```
System.out.println(10/6); // == 1
```

Guess the output #2

```
char x = 'X';  
int i = 0;  
System.out.print(true ? x : 0);  
System.out.print(true ? x : i);
```

Guess the output #2

```
char x = 'X';  
int i = 0;  
System.out.print(true ? x : 0); // == X  
System.out.print(true ? x : i); // == 88
```

Guess the output #3

```
System.out.println(12345+54321);
```

Guess the output #3

```
System.out.println(12345+5432L); // == 17777
```


Guess the output #4

```
public boolean indecision() {  
    try {  
        throw new Exception();  
    }  
    finally {  
        return false;  
    }  
}
```

Guess the output #4

```
public boolean indecision() {  
    try {  
        throw new Exception();  
    }  
    finally {  
        return false; // always executed last!  
    }  
}
```

Guess the output #5

```
class Counter implements Runnable {
    static int i = 0;

    public void run() {
        i++;
        System.out.print(i + " ");
    }
}

for (int i = 0; i < 10; i++) {
    new Thread(new Counter()).start();
}
```

Guess the output #5

```
class Counter implements Runnable {
    static int i = 0;

    public void run() {
        i++;
        System.out.print(i + " "); // 1 2
    }
}

for (int i = 0; i < 10; i++) {
    new Thread(new Counter()).start();
}
```

Guess the output #5

```
class Counter implements Runnable {
    static int i = 0;

    public void run() {
        i++;
        System.out.print(i + " "); // 1 2 4 4 ...
    }
}

for (int i = 0; i < 10; i++) {
    new Thread(new Counter()).start();
}
```

Questions?

jf@zeroturnaround.com