

Overview

Getting started

How much time to invest?



Mean **weekly** effort:

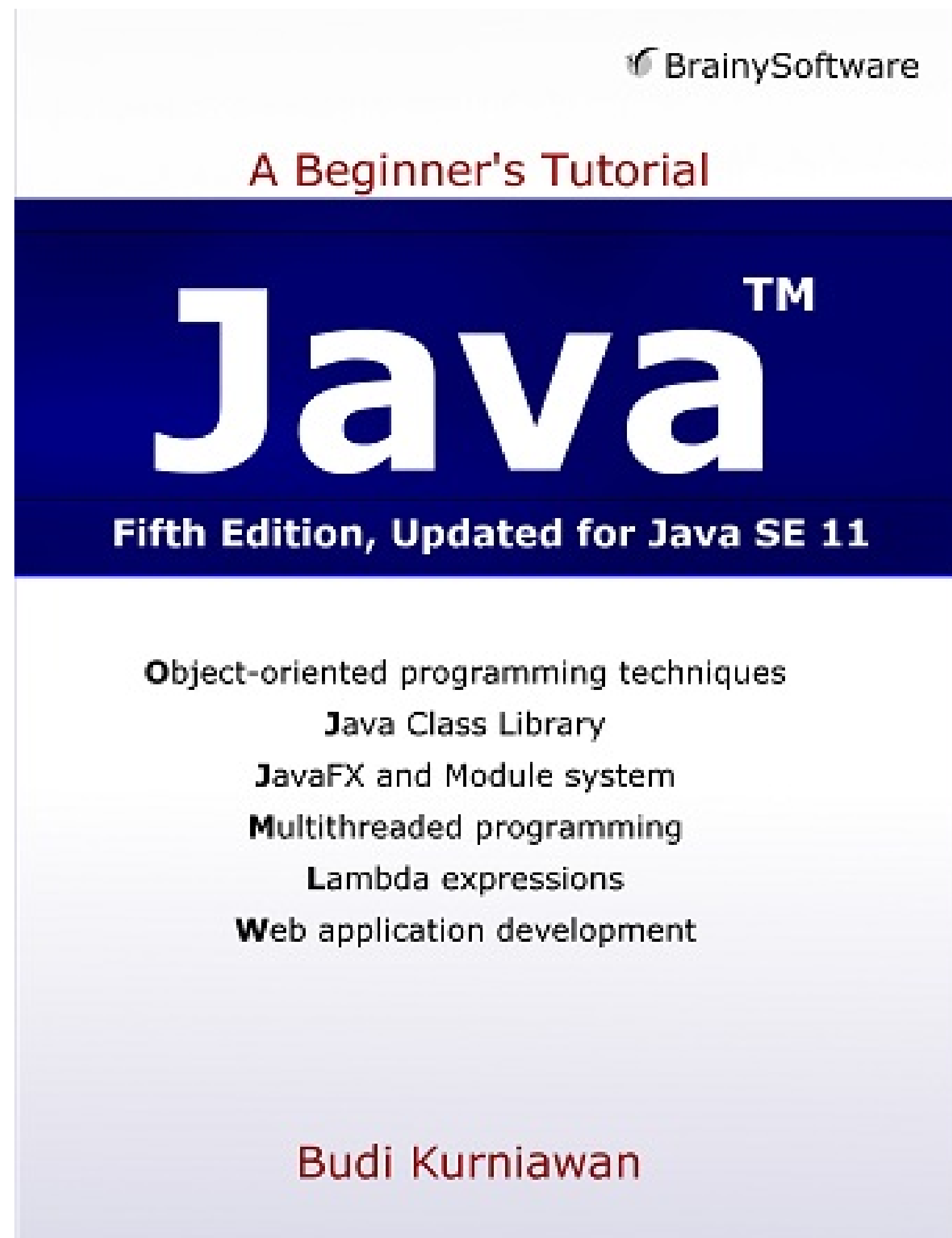
- 40 hours (60 mins) workload / week
- 30 ECTS for all current lectures
- $40h * 8 / 30 \Rightarrow$ **10 hours 40 minutes**
- Lectures + Exercises:
8 x 45 minutes = 6 hours
- Supplementary weekly effort:
4 hours and 40 minutes

Overview

Getting started

⇒ Lecture related resources

Recommended reading resources I



Recommended reading resources II

Primary

- [Java: A Beginner's Tutorial \(6th Edition\)](#)

Secondary

- [Java ist auch eine Insel](#) of 15-th edition book 2020 including [Java™ 14](#).
- [Grundkurs programmieren in Java](#)

Your biggest enemies

Discussion boards



Push news

Entertainment

Messenger(s)

»Social« networks



Study time?

**Work-life-
balance?**

Aus „Der Postillion“ :

*Mann, der am Handy
nur mal eben die Uhrzeit nachschauen wollte,
chattet acht Minuten auf WhatsApp,
schaut drei YouTube-Videos
und liest einen Artikel über Peru,
weiß aber am Ende immer noch nicht, wie spät es ist*

4 most imperative study objectives

1. **MANAGE YOUR TIME!**
2. **MANAGE YOUR TIME!**
3. **MANAGE YOUR TIME!**
4. **MANAGE YOUR TIME!**

Bradley Kjell: Introduction to Computer Science using Java

- German translation by Heinrich Gailer

Udemy: Java Tutorial for Complete Beginners

Video tutorials and related source code examples. Registration required.

Unix and the terminal

- [The Unix Shell](#) / Software-carpentry, nice video collection. Each section is also available in PDF and PowerPoint™ format.
- [UNIX Tutorial for Beginners](#), text oriented.
- [Introduction to Unix commands](#)

Online programming, automated feedback

<http://codingbat.com>

No registration required.

<https://www.programmr.com/zone/java>

- Hunt for “Challenges” within page.
- Registration required.

<https://www.codewars.com>

- Registration or github.com login required.

Online programming I

codeabbey.com

Problem list.

rosettacode.org

Programming tasks (including solutions for multiple languages).

reddit.com

Daily Programmer.

Project Euler

Registration required for keeping track of your exercises' status. The following exercises in particular are considered to be useful with respect to this lecture:

[1](#), [2](#), [4](#), [5](#), [8](#), [9](#), [11](#).

Java Programming Tutorial

Basic and more difficult exercises

Java Programming Exercises

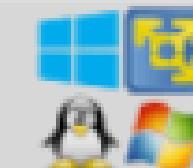
Start from the [easier exercises](#).

Choosing the SD1 bwLehrpool VM



Natives Linux

Zuletzt verwendet



Alle Kurse

▼ Kursumgebungen speziell für diesen Raum



113219 Linux Essentials



Anwendungssicherheit



E-Exam 113105 Softwareentwicklung 1 freies Üben



E-Exam 119203a Mensch-Computer-Interaktion (HCI) Probeklausur



IW Prüfung



Web Development Enterprise

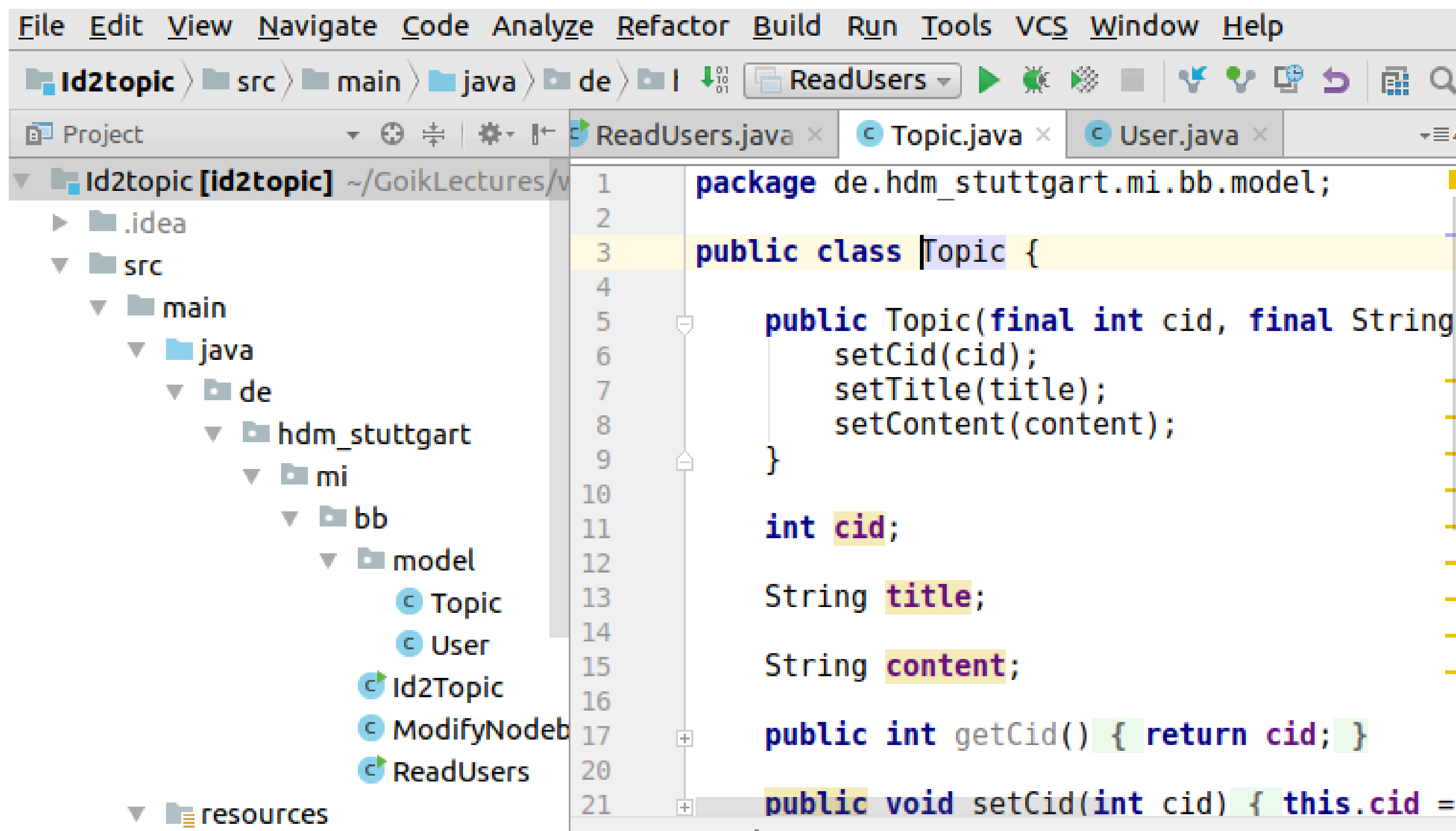
▼ Allgemeine Kursumgebungen



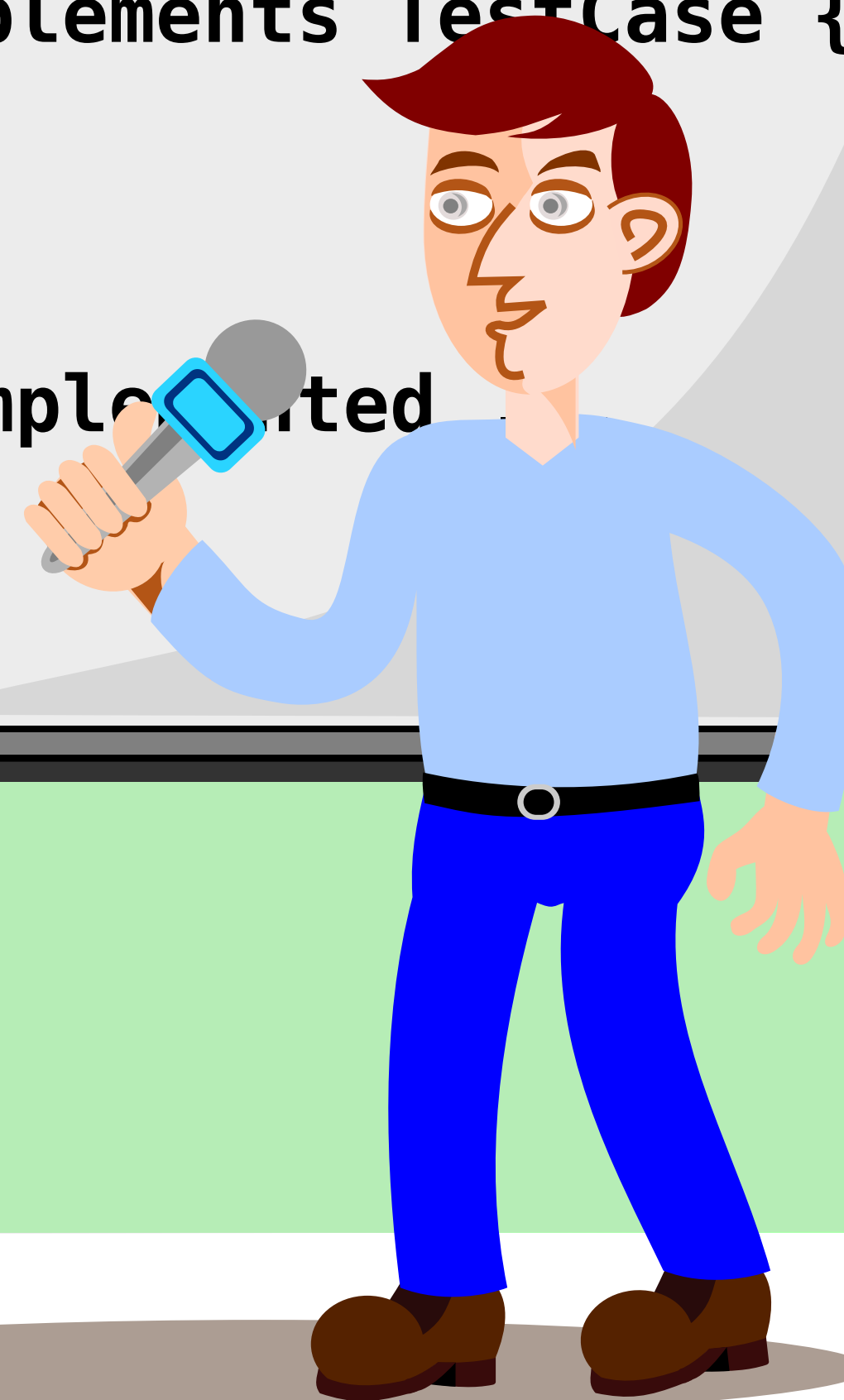
113105 Softwareentwicklung 1



113118a Grundlagen der Mediengestaltung




```
public class Test implements TestCase {  
    void tearDown() {  
        init();  
    }  
    // Methods to be implemented  
}
```



Virtualbox / VMware player based virtual Linux image

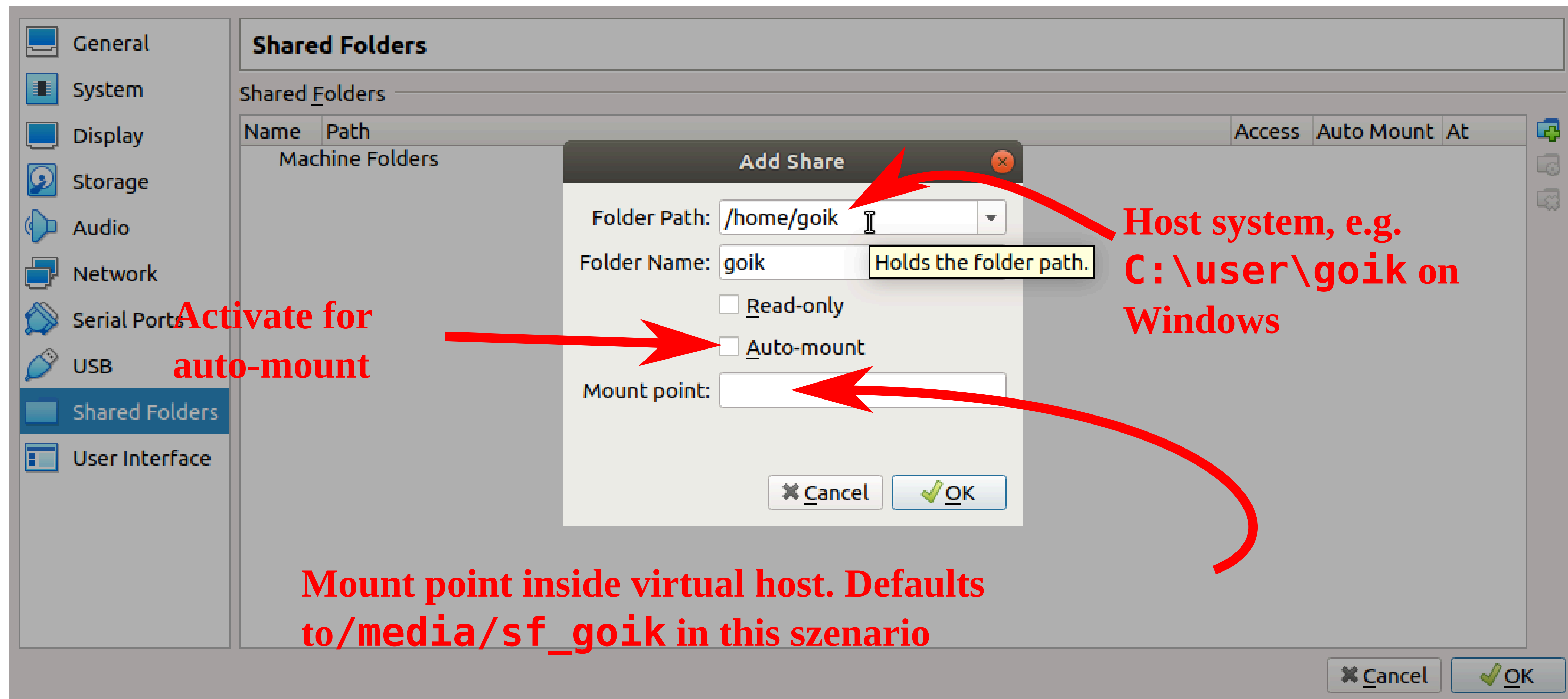
- Contains all MI pool workstation Linux software.
- Available for free [VirtualBox](#) desktop as [compressed image](#).
- The beast is quite big (~20 GB on disk, ~ 5GB compressed download provided as [rar](#) chunks)!

Tip

You may favour a wired connection over WIFI !

- Alternative: Native or dual boot [Ubuntu](#) “[Desktop](#)” installation.

Virtualbox™ settings



Overview

Getting started

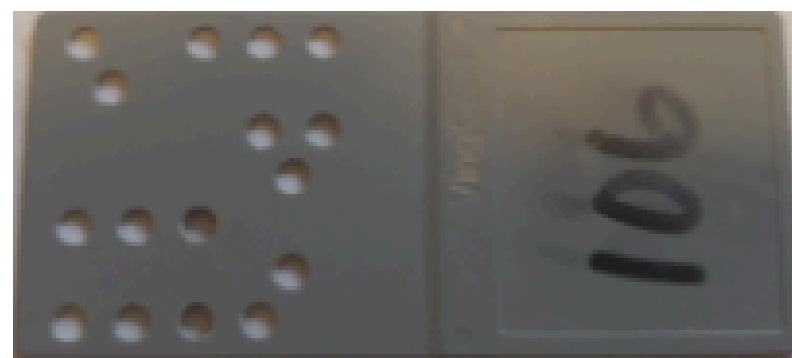
⇒ Using the exercises

Embedded exercises

- Tight relationship to the E-examination.
- Complete [list of exercises on offer](#).

Hotel key cards

1 Q: A hotel supplies the following type of cards for opening room doors:



A customer is worried concerning the impact of loosing his card. For security eventually run short on available combinations.

Discuss this argument by estimating the number of distinct patterns.

2 A:  open solution (Did you try hard enough yourself?)

3  [Create comment](#)

Overview

Getting started

➡ Related resources

Either of:

- Read your mails at <https://ox.hdm-stuttgart.de> regularly.
- or
- Activate mail forwarding from <https://ox.hdm-stuttgart.de> to your “real” email account.

Configure MI [VPN](#) client access

- External [MI E-examination system](#) access requires [VPN](#):
 - Past years' E-examinations.
 - Your personal exam results.
- OpenVPN wiki [installation page](#) (Login required).
- HdM_MI_stud.ovpn allows for using a maximum of MI services.

- <https://cloud.mi.hdm-stuttgart.de>.
- 25 GB free disk space.
- Desktop and mobile clients.



NSA free
(Maybe)

MI File server

- Accessing your computer pool home directory.
- Domain **hdm**, requires username **hdm**\xy123 for login
- Windows share \\mi-ad1.srv.mi.hdm-stuttgart.de\xy123 or \\192.168.111.15\xy123.
- Requires **Mi VPN**.

- <https://gitlab.mi.hdm-stuttgart.de>.
- Collaborative software development.

Getting started

→ Coached exercises

Coached exercises

- Tuesday and Wednesday 17:45-19:15.
- Seminar groups of ~12 participants assigned to a tutor.

Bonus points

- Precondition: You must pass the examination based on its own score excluding bonus points.
- Examination: E.g. 90 points / 100% resulting in “1,0”, 45 points / 50% resulting in “4.0”.
- 0-10 bonus points on top of examination score in case of reaching at least 50% examination points.
- Examples:
 - 40 examination points: “Failed” regardless of any number of bonus points
 - 45 Examination points, 10 bonus points. Result: 55 points resulting in a 3.0 mark rather than 4.0.

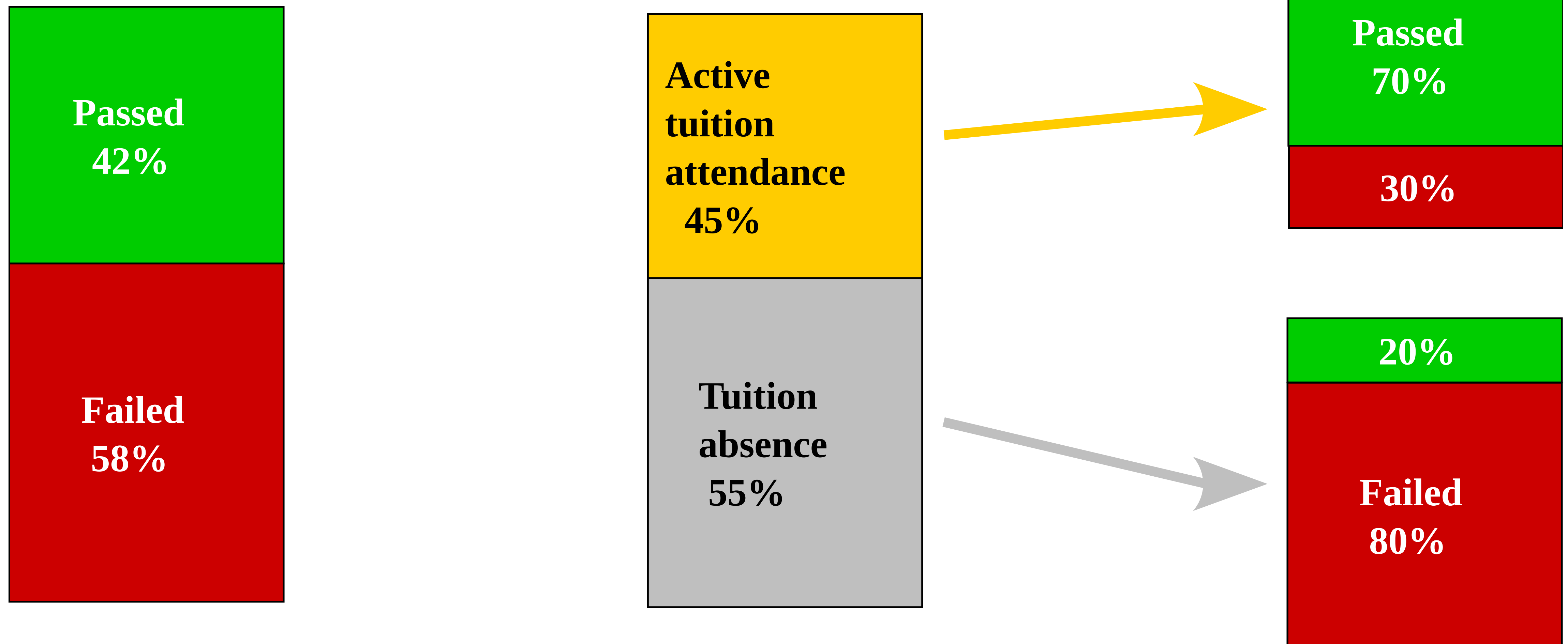
Seminar rules and bonus points

- Join exactly one group at the [MI E-learning system](#).
- Bonus point requirements:
 1. 80% participation rate of all weekly appointments
 2. Presenting **at least three exercise** solutions of <https://freedocs.mi.hdm-stuttgart.de/apb.html>.

Presenting exercise solutions

- Give a brief account of the exercise in question.
- Explain your solution's concept and present your code.
- Explain possible problems / pitfalls.
- Ask your tutor for exercises to avoid thematic clashes

Tuition attendance and exam results



Getting started

⇒ Using plain **Java™**

Edit - compile - execute

Source code: **HelloWorld.java**

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Hello, world");  
    }  
}
```

compile

Java Compiler

javac

Bytecode:

HelloWorld.class

execute

Java Virtual Machine (JVM)

java

Bytecode
verifier

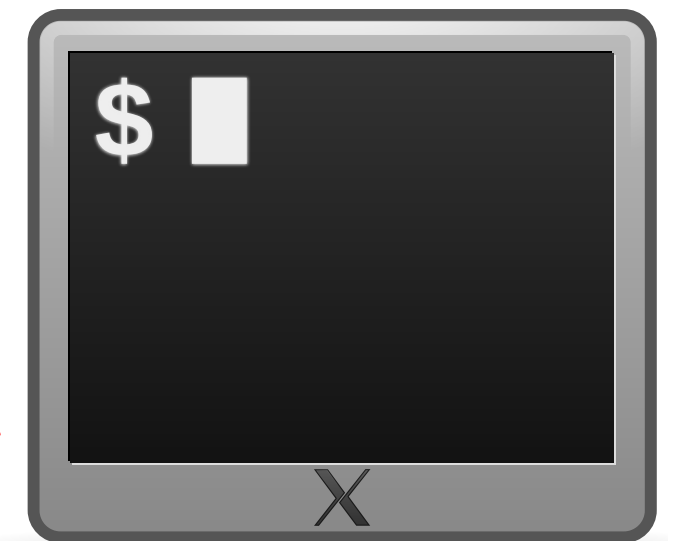
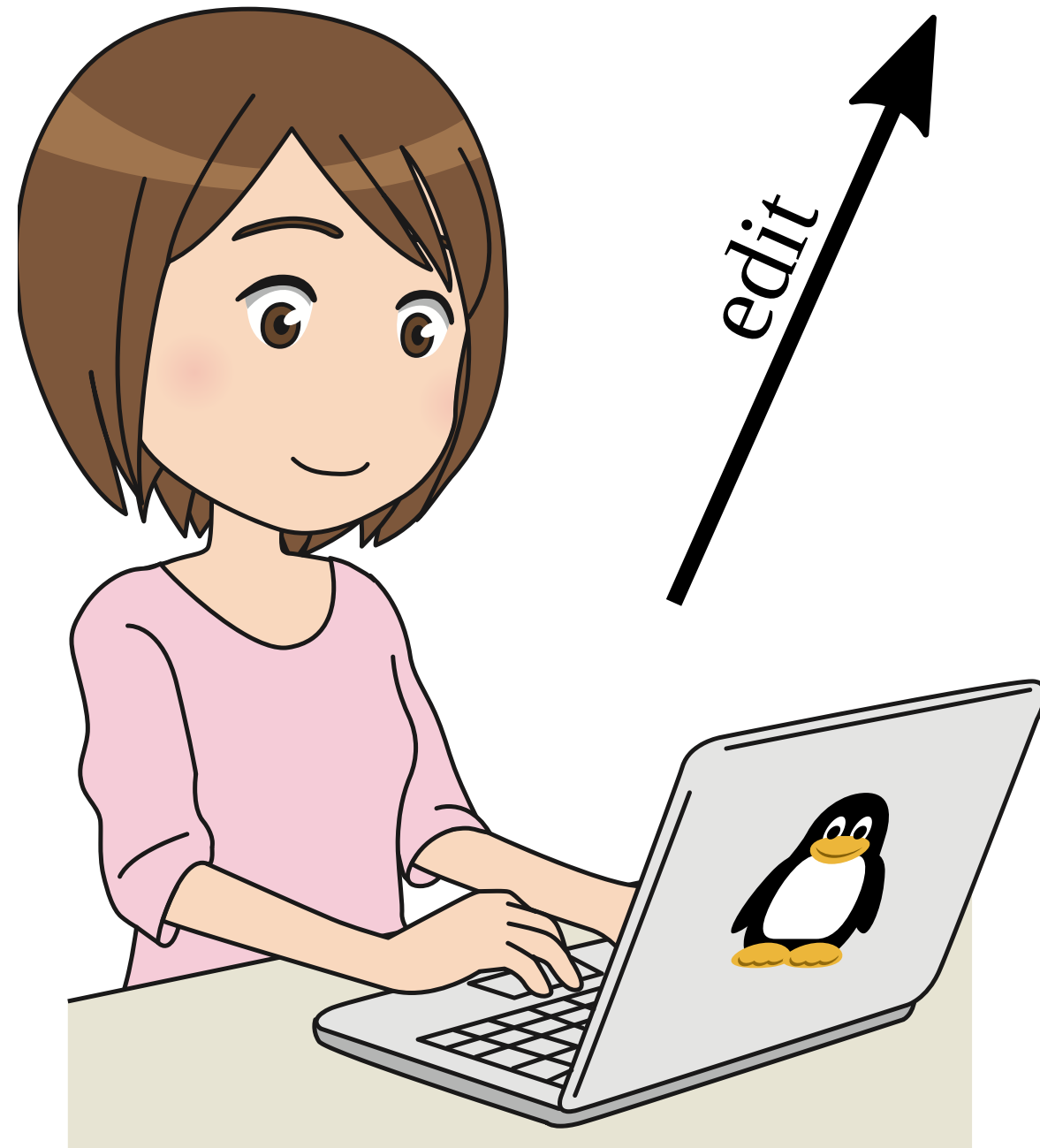
Classloader

Java Runtime
System

Native
OS

Standard
output

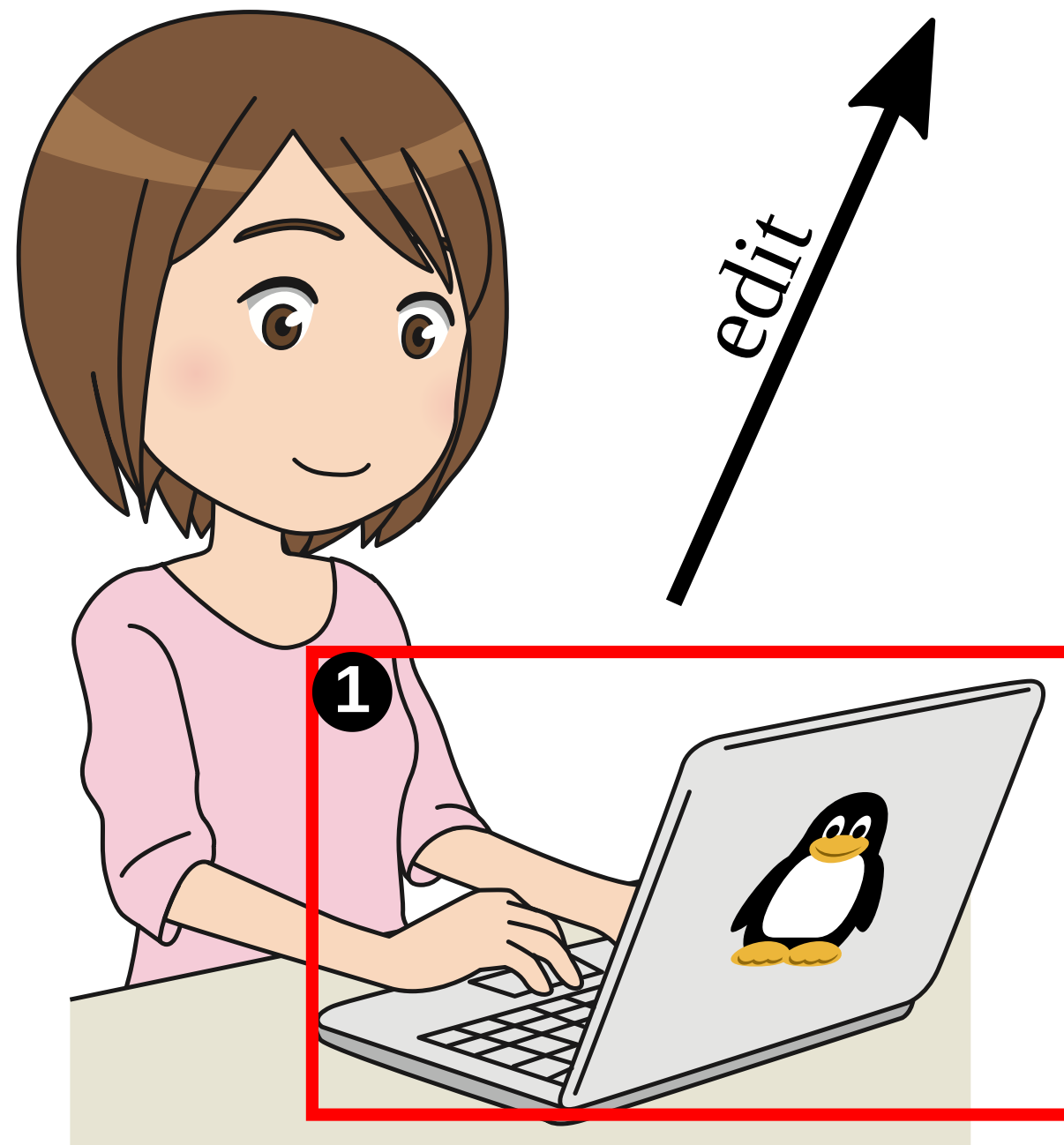
Standard
Input



2

Source code: **HelloWorld.java**

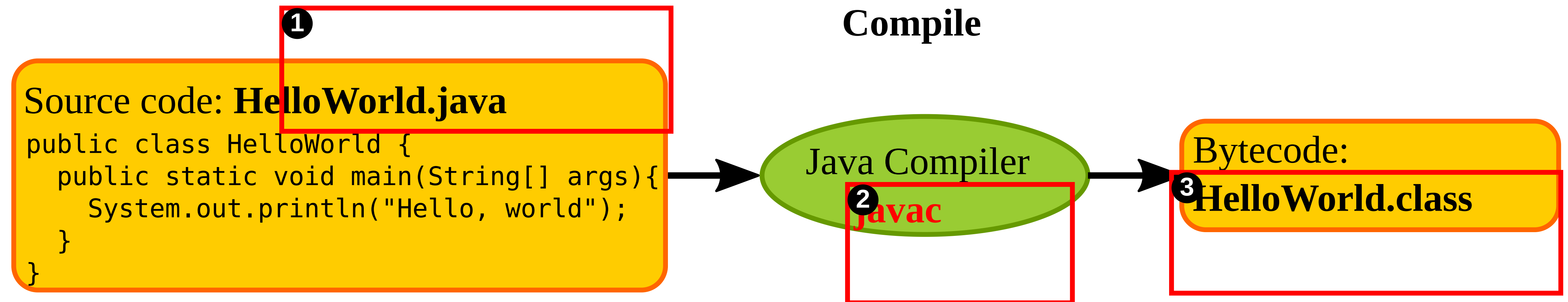
```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Hello, world");  
    }  
}
```



Defining class HelloWorld

```
// Filename HelloWorld.java ❶  
  
public class HelloWorld ❷ {  
  
    public static void main(String[] args) ❸ {  
        System.out.println("Hello, world"); ❹  
    }  
  
}
```


Compiling Java™ file



Command line Java™ file compilation

```
~/tmp$ ls -al HelloWorld.class
ls: cannot access 'HelloWorld.class': No such file or directory

~/tmp$ javac HelloWorld.java

~/tmp$ ls -al HelloWorld.class
-rw-r--r-- 1 goik fb1prof 419 Sep 23 15:44 HelloWorld.class
```

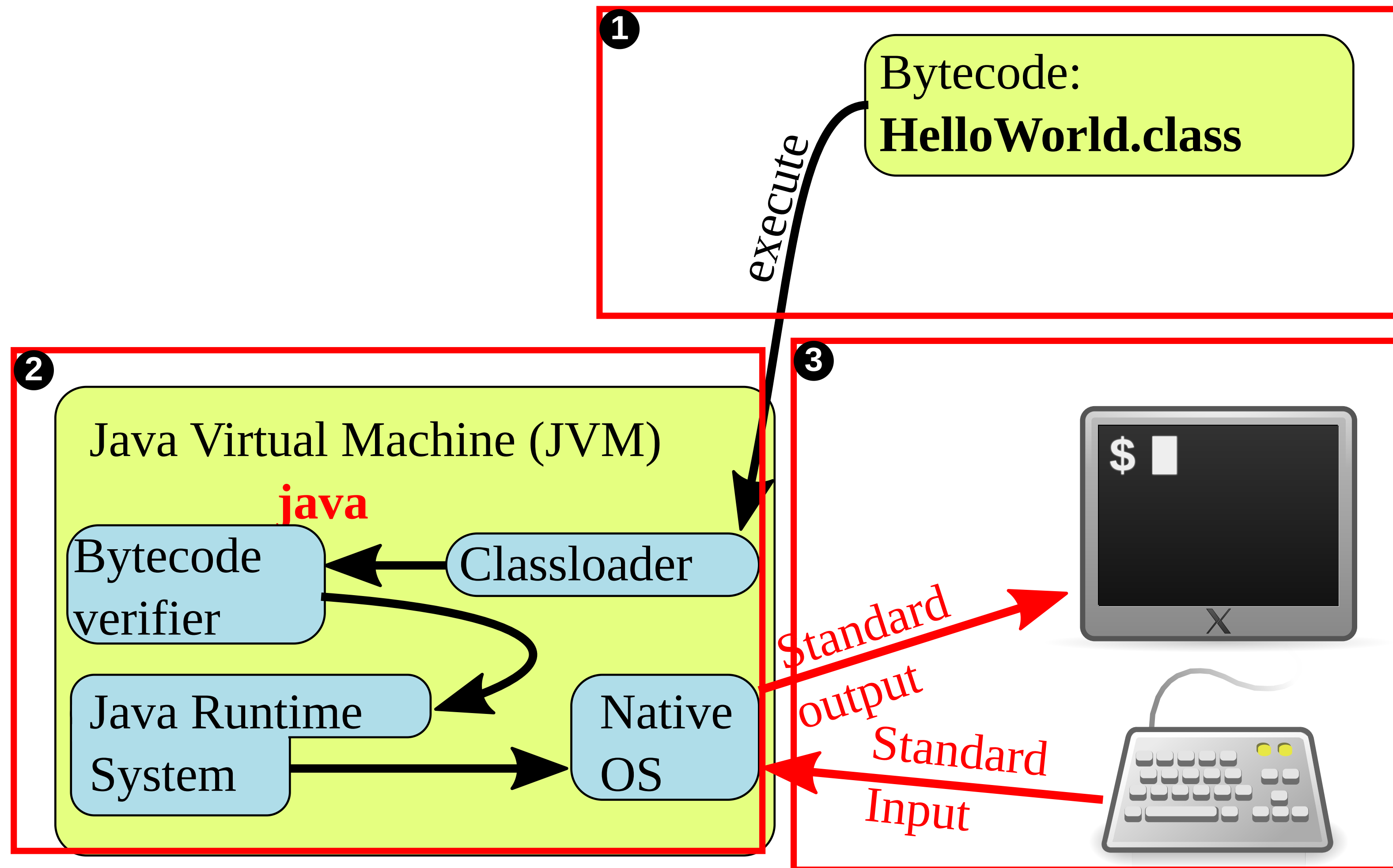

Java byte code file HelloWorld.class

```
Ëp¼^@^@6^@^]
^@^F^@^O      ^@^P^@^Q^H^@^R
^@^S^@^T^G^@^U^G^@^V^A^@^F<init>^A^@^C()V^A^@^DCode^A^@^OLineNumberTable^A^@^Dmain^A^@^V(
  [Ljava/lang/String;)V^A^@
SourceFile^A^@^0HelloWorld.java^L^@^G^@^H^G^@^W^L^@^X^@^Y^A^@^LHello, world^G^@^Z^L^@^[^@\\
^\\^A^@
HelloWorld^A^@^Pjava/lang/Object^A^@^Pjava/lang/System^A^@^Cout^A^@^ULjava/io/PrintStream;\\
^A^@^Sjava/io/PrintStream^A^@^Gprintln^A^@^U(Ljava/lang/String;)V^@!^@^E^@^F^@^@^@^@^@^\\
B^@^A^@^G^@^H^@^A^@      ^@^@^@^@]^@^A^@^A^@^@^@^@^@^@E*.^@^A±^@^@^@^@^@^A^@
^@^@^@^@^@^@^@^@^@B^@      ^@^K^@^L^@^A^@      ^@^@^@%^@^B^@^A^@^@^@      ²^@^B^R^C¶^@^D±^@^@^@^@^@^A^@
^@^@^@
^@^B^@^@^@^@^D^@^H^@^E^@^A^@^M^@^@^@^@^B^@^N
```

Source code vs. bytecode

HelloWorld.java	HelloWorld.class
• Human readable (kind of 😄). • High abstraction level. • Text file	• Machine readable instructions. • Non-editable (usually). • Binary file.

Executing byte code file HelloWorld.class



Command line byte code file HelloWorld.class execution

```
> java HelloWorld  
Hello, world
```

Remark: This executes HelloWorld.class rather than HelloWorld.java.

Getting started

⇒ Using **IntelliJ IDEA**

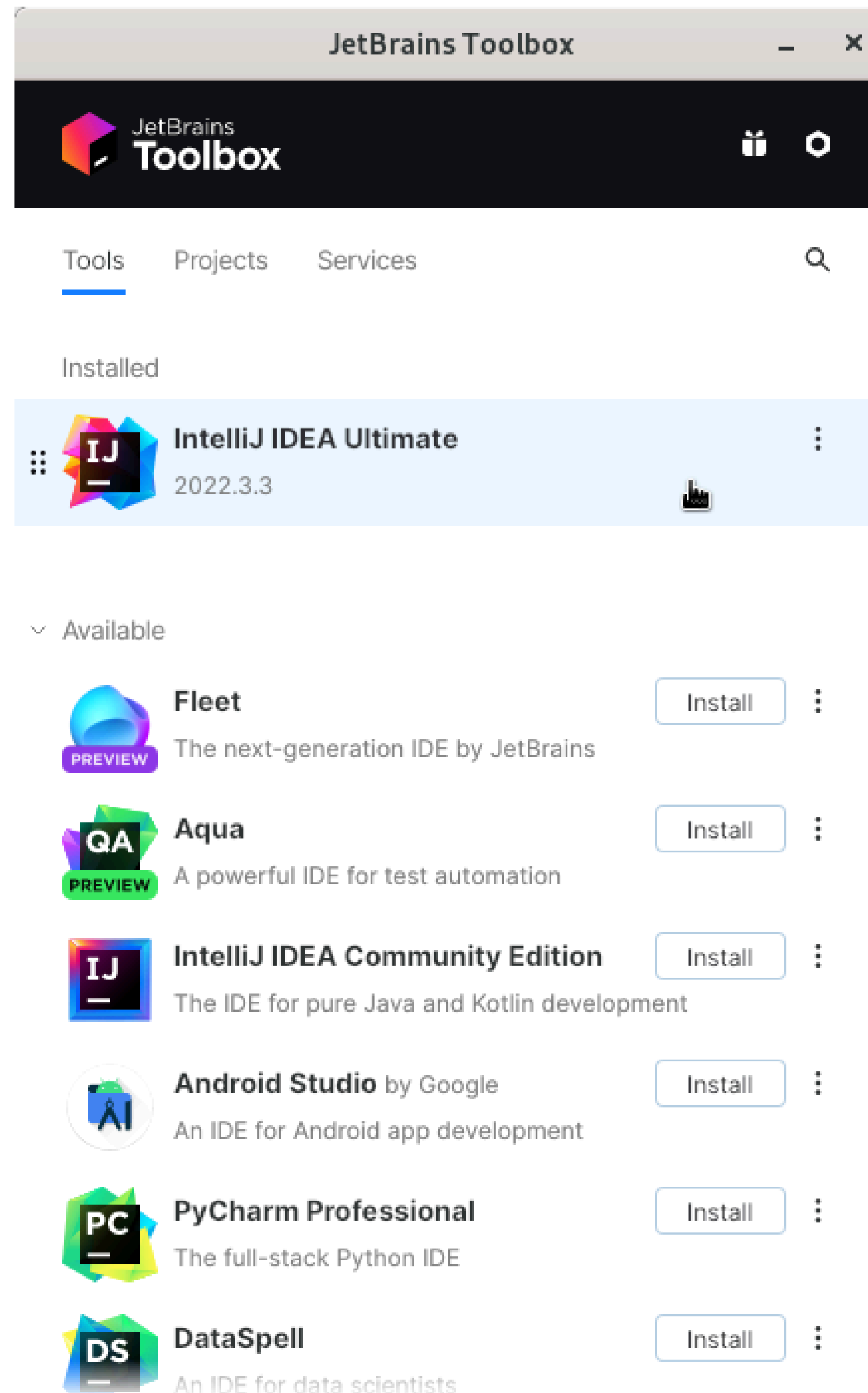
Prefer a **L**ong **T**erm **R**elease (**LTS**) e.g. 17, 21, ...

- Windows / Mac-OS: Manual [Oracle](#) or [OpenJDK](#) installation.
- Linux: Package install

Debian / Ubuntu	Fedora / Redhat
<pre>apt install openjdk-21-jdk</pre>	<pre>dnf install java-21-openjdk-devel</pre>

IntelliJ IDEA installation

- IntelliJ IDEA Toolbox based installation
- Choose “Ultimate”.



Idea "Ultimate" license types

- Activation code for offline usage, apply at:

www.jetbrains.com/shop/eform/students

- Using HdM/MI license server:

http://jetbrains.mi.hdm-stuttgart.de:11111

See [MI wiki](#) for further details.

Alternative: Using the HdM license server

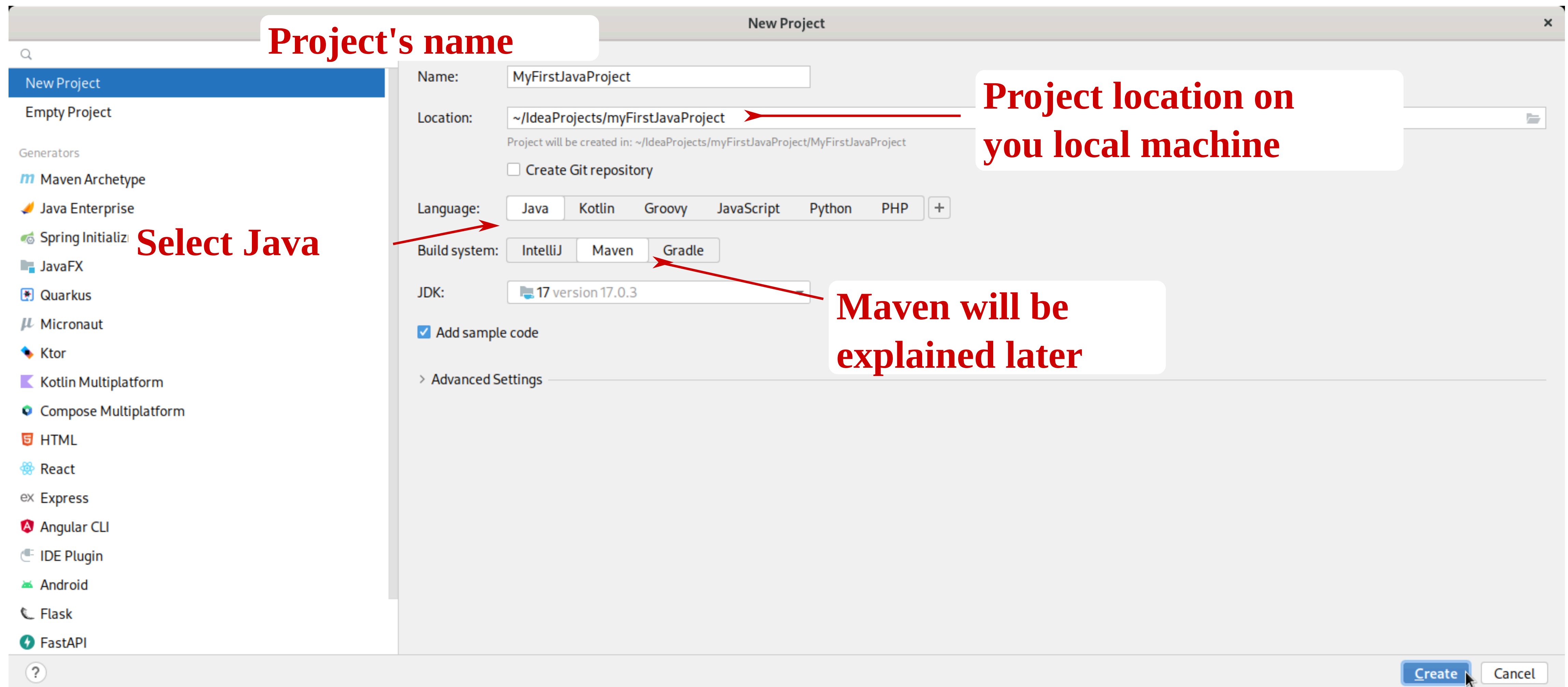


Insert address:

`http://jetbrains.mi.hdm-stuttgart.de:11111`

External usage requires **VPN** !

Creating a new Java project



Getting started
⇒ Play!

Getting first Java™ impressions

- Copy code you probably do not (yet) completely understand
- Try to guess what's going on
- Execute and watch the outcome
- Add minor modifications and re-execute.
- Don't worry: You'll get a full understanding later. (Promised! 🙄)

Followup exercises

1. Extending class HelloWorld
2. Working with variables
3. A conditional
4. A loop