

# How much time to invest?

---

Mean **weekly** effort:



# How much time to invest?

---

Mean **weekly** effort:

•40 hours (60 mins) workload / week



# How much time to invest?

---



## Mean **weekly** effort:

- 40 hours (60 mins) workload / week
- 30 ECTS for all current lectures

# How much time to invest?

---



## Mean **weekly** effort:

- 40 hours (60 mins) workload / week
- 30 ECTS for all current lectures
- $40\text{h} * 8 / 30 \Rightarrow$  **10 hours 40 minutes**

# 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:  
 $8 \times 45 \text{ minutes} = 6 \text{ hours}$

# 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**

# Recommended reading resources I

---

BrainySoftware

A Beginner's Tutorial

# Java™

Fifth Edition, Updated for Java SE 11

Object-oriented programming techniques

Java Class Library

JavaFX and Module system

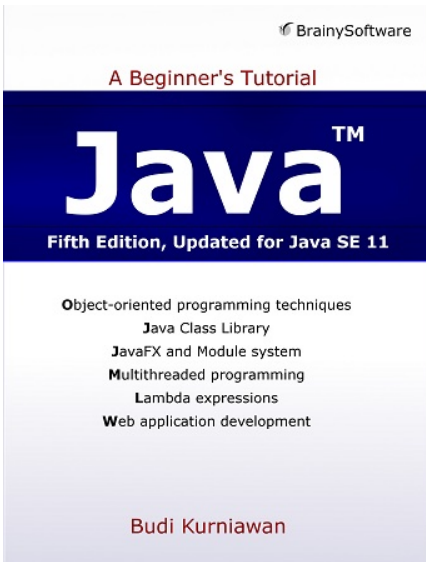
Multithreaded programming

Lambda expressions

Web application development

Budi Kurniawan

# Recommended reading resources I





# Recommended reading resources I

BrainySoftware

A Beginner's Tutorial

# Java™

Fifth Edition, Updated for Java SE 11

- Object-oriented programming techniques
- Java Class Library
- JavaFX and Module system
- Multithreaded programming
- Lambda expressions
- Web application development

Budi Kurniawan

Christian Ullenboom

# Java

ist auch eine Insel

Einführung, Ausbildung, Praxis

Erste Insel

- Programmieren mit der Java Platform, Standard Edition 9
- Java von A bis Z: Einführung, Praxis, Referenz
- Von Ausdrücken und Anweisungen zu Klassen und Objekten

Aktuell zu Java 9

13., aktualisierte und überarbeitete Auflage

Rheinwerk Computing

Über 100.000 begeisterte Leser

GRUNKURS PROGRAMMIEREN IN JAVA

HANSER

AN ALTERNATIVE SOFTWARE UND TOOLS, ALS BEWEISFÜR EIN NEUES PROGRAMMIERPARADIGMA, LÖSUNGEN & MEHR

# 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

---

# Your biggest enemies

---



# Your biggest enemies

---

## Discussion boards



# Your biggest enemies

---

**Discussion boards**



**Push news**

# Your biggest enemies

---

**Discussion boards**



**Push news**

**Entertainment**

# Your biggest enemies

---

## Discussion boards



**Push news**

**Entertainment**

**Messenger(s)**



# Your biggest enemies

---

## Discussion boards



**Push news**

**Entertainment**

**Messenger(s)**

**»Social« networks**

# Your biggest enemies

---

## Discussion boards

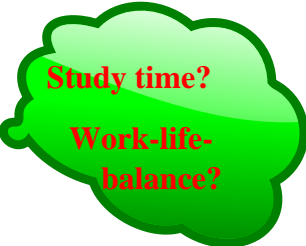


Push news

Entertainment

Messenger(s)

»Social« networks



Study time?

Work-life-balance?

# German humour

---

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!

# Online tutorials

---

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](https://www.programmr.com/zone/java)

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

[https://  
www.codewars.com](https://www.codewars.com)

- Registration or github.com login required.

# Online programming I

---

[codeabbey.com](https://codeabbey.com)

Problem list.

[rosettacode.org](https://rosettacode.org)

Programming tasks (including solutions for multiple languages).

[reddit.com](https://reddit.com)

Daily Programmer.



# Online programming II

---

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.

# Openjdk source code repository

---

- Welcome to the JDK!

# Java Visualizer

---



Java Visualizer

(beta: [report a bug](#))

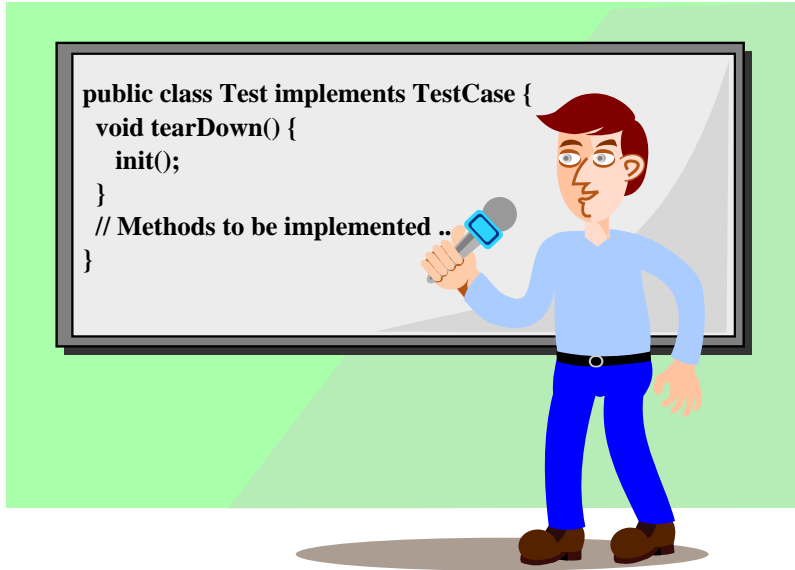


Write your Java code here:

```
1 public class ClassNameHere {  
2     public static void main(String[] args) {  
3  
4     }  
5 }
```

# Live lecture additions

---



# Remote lecture participation

The screenshot shows a web browser window with a BigBlueButton meeting. The browser address bar shows the URL: `bbb-03.hdm-stuttgart.de/html5client/join?sessionToken=wq5bbrehxd5k2asv`. The interface is split into a chat window on the left and a presentation slide on the right.

**Chat Window (Left):**

- MESSAGES:** Public Chat
- NOTES:** Shared Notes
- USERS (2):** Dr. Martin Goik (You), Joe Foo
- Message 1:** Willkommen in der SD1-Konferenz!  
Verwenden Sie bitte ein Headset, um Störungen zu vermeiden.
- Message 2:** To invite someone to the meeting, send them this link:  
<https://konferenz1.hdm-stuttgart.de/b/drm-yj3-aex-m29>
- Message 3:** Joe Foo 2:36 PM: Hi there, I just joined in
- Message 4:** Dr. Martin Goik 2:36 PM: Welcome

**Presentation Slide (Right):**

- Title:** Willkommen in BigBlueBut
- Section:** Tipps für ein erfolgreiches Meeting:
- Tip 1:** Kopfhörer oder Headset verwenden → Bessere Tonqualität & vermeidet Rückkopplung. (Illustrated with a headset icon)
- Tip 2:** Audio und Video im Meeting steuern
- Audio Controls:** Three circular icons: a microphone (labeled "Sprechen" with "Mikrofon an/stumm" below), a speaker (labeled "Audio" with "Mikrofon und Lautsprecher wählen" below), and a video camera (labeled "Video" with "Webcam ein/a virtuelle Hintergründe" below).

# Virtualbox / VMware player based virtualized Linux image

---

- Contain all MI pool workstation Linux software.
- Available for free VMware Workstation Player as compressed image. Apple users: Consider buying VMWare Fusion.
- Available for free VirtualBox desktop virtualization as compressed image.
- The beasts are quite big (~20 GB on disk, ~ 5GB compressed download)! **You may prefer a wired connection in favour of WiFi !**
- Alternative: Native or dual boot Ubuntu “Desktop” installation.

# Virtualbox™ settings

File Machine Help

**Tools**

New Settings Discard Start

**Windowf**  
Saved

**Mibuntu**  
Powered Off

**mi\_public**  
Powered Off

**General**

Name: mi\_public  
Operating System: Ubuntu (64-bit)  
Settings File Location: /ma/goik/VirtualBox VMs/mi\_public


**System**

Base Memory: 6850 MB  
Boot Order: Floppy, Optical, Hard Disk  
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

**Display**

Video Memory: 128 MB  
Graphics Controller: VMSVGA  
Remote Desktop Server: Disabled  
Recording: Disabled

**Preview**



# Virtualbox™ settings

The screenshot shows the VirtualBox settings window for a VM named 'mi\_public'. The interface includes a menu bar (File, Machine, Help), a toolbar (New, Settings, Discard, Start), and a left sidebar with a list of VMs: 'Windoof', 'Mibuntu', and 'mi\_public'. The 'mi\_public' VM is selected and highlighted in blue. The main settings area is divided into three panes: 'General', 'System', and 'Display'. A red arrow points from the text 'Adapt your VM's settings' to the 'Settings' button in the toolbar.

**File** **Machine** **Help**

**Tools**

**Windoof** Saved

**Mibuntu** Powered Off

**mi\_public** Powered Off

**New** **Settings** **Discard** **Start**

**Adapt your VM's settings**

**General**

Name: mi\_public  
Operating System: Ubuntu (64-bit)  
Settings File Location: /ma/goik/VirtualBox VMS/mi\_public

**System**

Base Memory: 4096 MB  
Boot Order: Floppy, Optical, Hard Disk  
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

**Display**

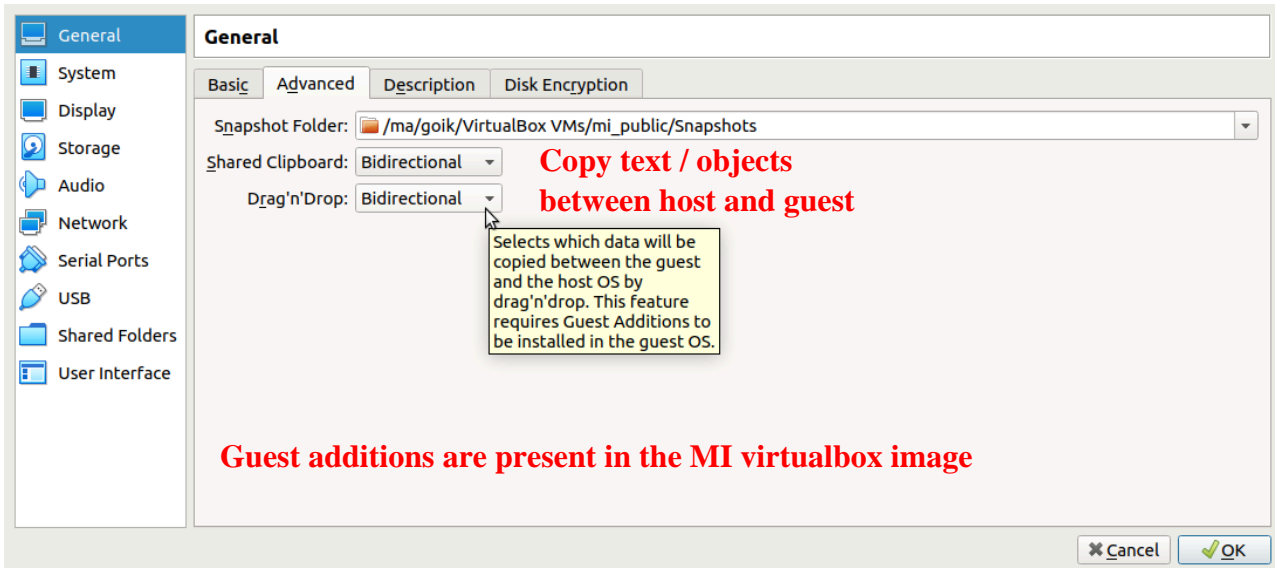
Video Memory: 128 MB  
Graphics Controller: VMSVGA  
Remote Desktop Server: Disabled  
Recording: Disabled

**Preview**

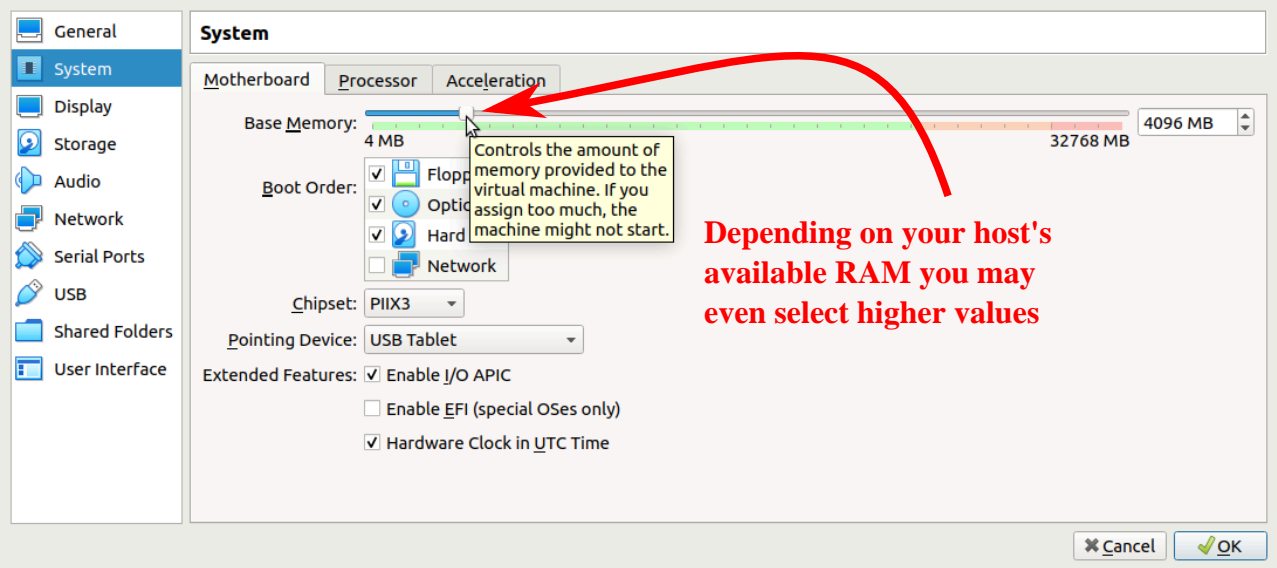
mi\_public



# Virtualbox™ settings



# Virtualbox™ settings



The screenshot shows the VirtualBox System settings window. The left sidebar lists various settings categories: General, System (selected), Display, Storage, Audio, Network, Serial Ports, USB, Shared Folders, and User Interface. The main window is titled 'System' and has three tabs: 'Motherboard', 'Processor', and 'Acceleration'. The 'Motherboard' tab is active, showing the 'Base Memory' slider set to 4096 MB. A tooltip explains that this slider controls the amount of memory provided to the virtual machine, warning that assigning too much memory might prevent the machine from starting. A red arrow points from the red text box to the slider. The red text box states: 'Depending on your host's available RAM you may even select higher values'. Other settings include 'Boot Order' (Floppy, Optical, Hard, Network), 'Chipset' (PIIX3), 'Pointing Device' (USB Tablet), and 'Extended Features' (Enable I/O APIC, Enable EFI, Hardware Clock in UTC Time).

**Depending on your host's available RAM you may even select higher values**

# Virtualbox™ settings

**Display** Just choose the maximum on offer

Screen Remote Display Recording

Video Memory: 0 MB 128 MB

Monitor Count: 1

Scale Factor: All Monitors Min Max

Graphics Controller: VM SVGA

Acceleration:  Enable 3D Acceleration  
 Enable 2D Video Acceleration

Cancel OK

# Virtualbox™ settings

The screenshot shows the 'Network' settings window for a virtual machine. The left sidebar contains various settings categories, with 'Network' selected. The main area is titled 'Network' and has tabs for 'Adapter 1', 'Adapter 2', 'Adapter 3', and 'Adapter 4'. Under 'Adapter 1', the 'Enable Network Adapter' checkbox is checked. The 'Attached to' dropdown is set to 'NAT'. The 'Name' field is empty. The 'Advanced' section is expanded, showing 'Adapter Type' set to 'Paravirtualized Network (virtio-net)'. The 'Promiscuous Mode' is set to 'Deny'. The 'MAC Address' is '08002758F011'. The 'Cable Connected' checkbox is checked, and the 'Port Forwarding' button is visible. A red arrow points from the text 'Better network performance' to the 'Adapter Type' dropdown. A yellow tooltip box explains that this dropdown selects the type of virtual network adapter, which affects the hardware provided to the VM. At the bottom right, there are 'Cancel' and 'OK' buttons.

**Better network performance**

Adapter 1 Adapter 2 Adapter 3 Adapter 4

Enable Network Adapter

Attached to: NAT

Name: \_\_\_\_\_

Advanced

Adapter Type: Paravirtualized Network (virtio-net)

Promiscuous Mode: Deny

MAC Address: 08002758F011

Cable Connected

Port Forwarding

Selects the type of the virtual network adapter. Depending on this value, VirtualBox will provide different network hardware to the virtual machine.

Cancel OK

# Virtualbox™ settings

**Shared Folders**

Shared\_Folders

Name	Path	Access	Auto Mount	At
Machine Folders				

**Add Share**

Folder Path: /home/goik *Host system, e.g. C:\user\goik on Windows*

Folder Name: goik *Holds the folder path.*

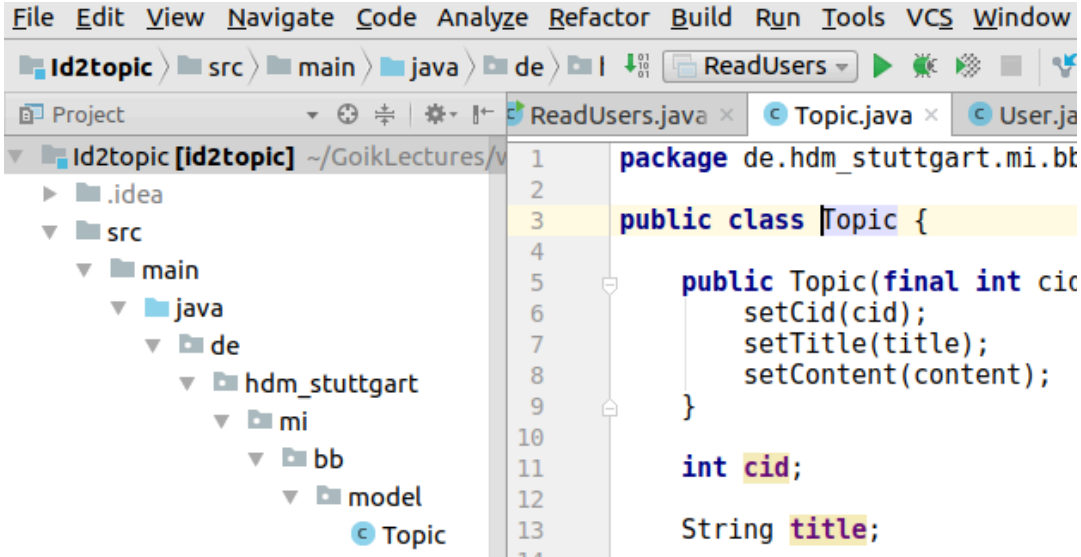
Read-only

**Auto-mount** *Activate for auto-mount*

Mount point: *Mount point inside virtual host. Defaults to /media/sf\_goik in this scenario*

Cancel OK

# IntelliJ IDEA IDE



The screenshot shows the IntelliJ IDEA IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, and Window. The breadcrumb navigation shows the path: Id2topic > src > main > java > de > hdm\_stuttgart > mi > bb > model > Topic. The Project tool window on the left displays the project structure, with the Topic class selected under the model package. The main editor window shows the source code of Topic.java, which includes a package declaration, a public class definition, a constructor, and two instance variables.

```
1 package de.hdm_stuttgart.mi.bb.model;
2
3 public class Topic {
4
5     public Topic(final int cid,
6                 final String title,
7                 final String content) {
8         setCid(cid);
9         setTitle(title);
10        setContent(content);
11    }
12
13    int cid;
14
15    String title;
```

# Embedded exercises

---

- Tight relationship to the E-examination.
- Complete list of exercises on offer.

# Using the exercises

---


## 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 losing 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](#)



# HdM mail server

---

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).
- HIMM\_stud. ovpn allows for using a maximum of MI services.

# MI Cloud server

---

- <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.
- Windows share \\mi-ad1.srv.mi.hdm-stuttgart.de\xy123 or \\192.168.111.15\xy123.
- Requires Mi VPN.

# MI Git versioning server

---

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

# 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>.



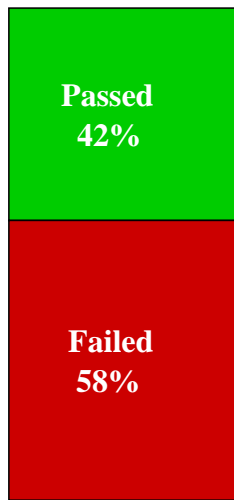
# Presenting exercise solutions

---

- 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

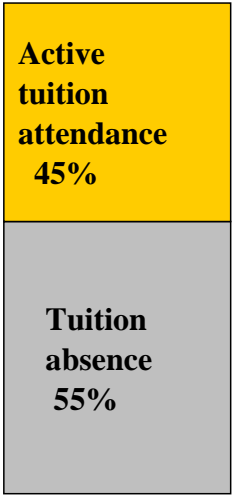
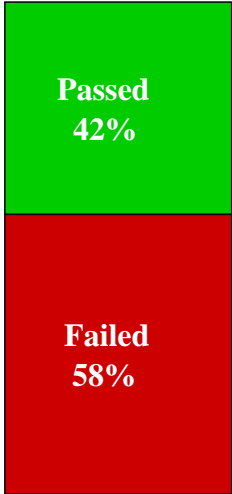
# Tuition attendance and exam results

---

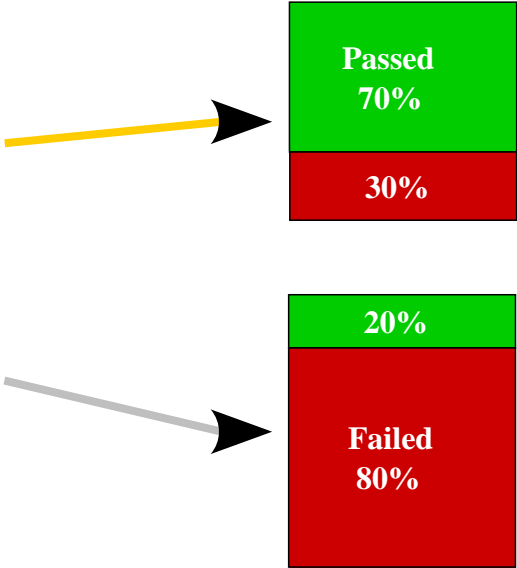
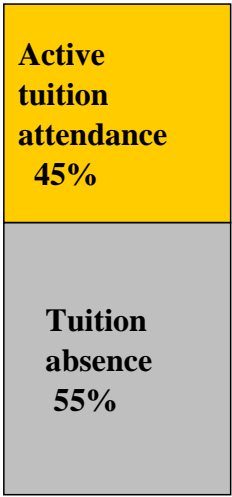
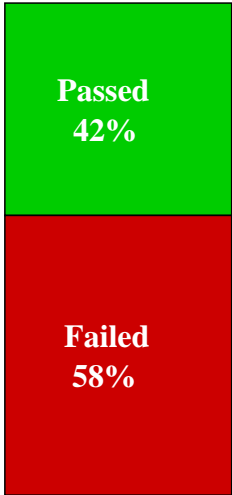


# Tuition attendance and exam results

---



# Tuition attendance and exam results



# Edit - compile - execute

---

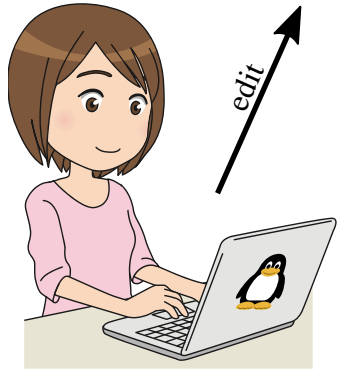


# Edit - compile - execute

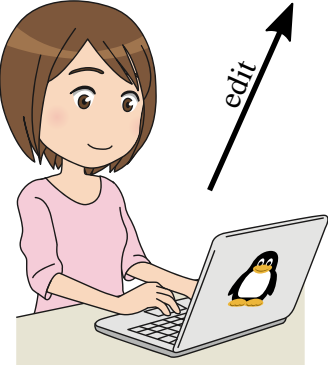
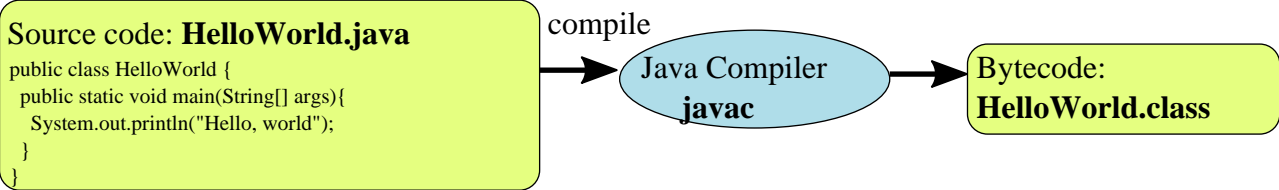
---

Source code: **HelloWorld.java**

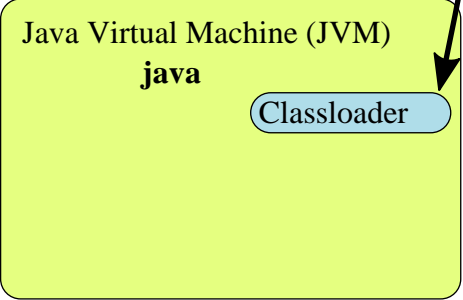
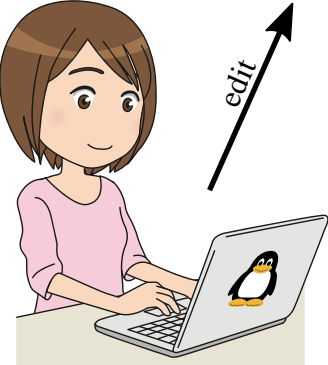
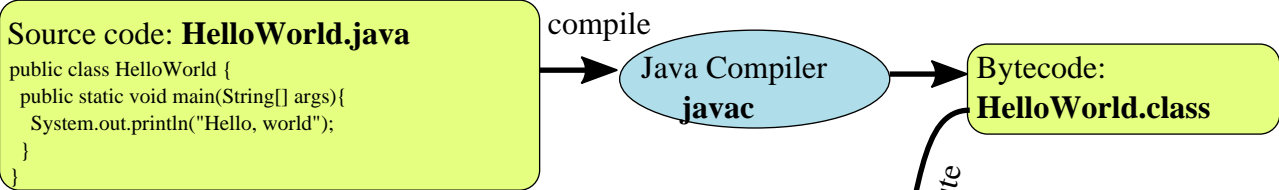
```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Hello, world");  
    }  
}
```



# Edit - compile - execute

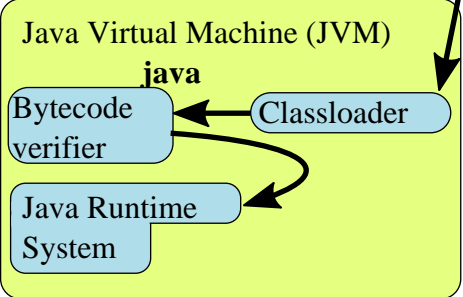
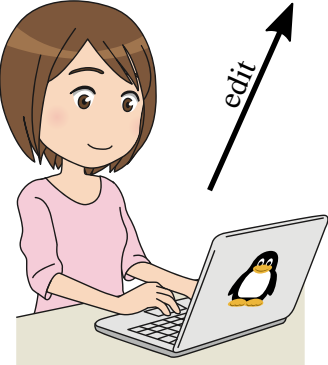
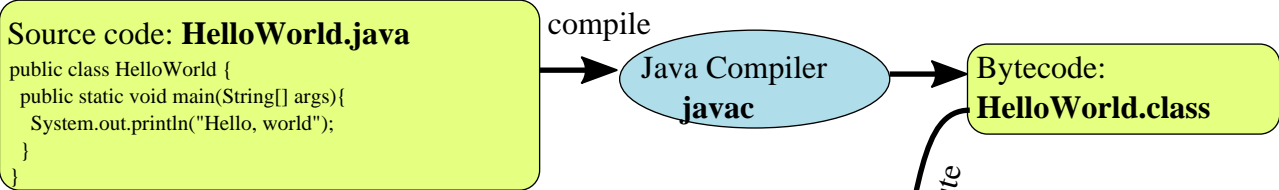


# Edit - compile - execute

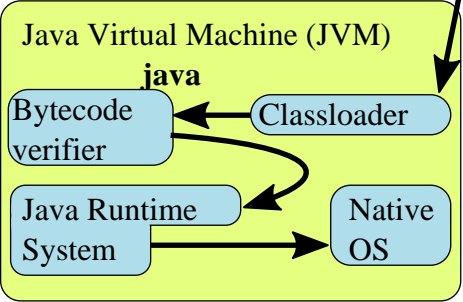
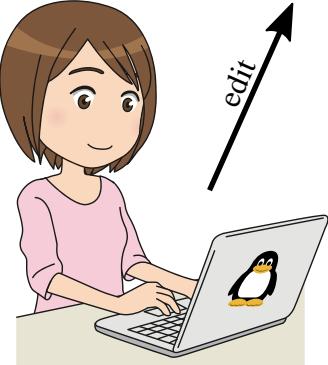
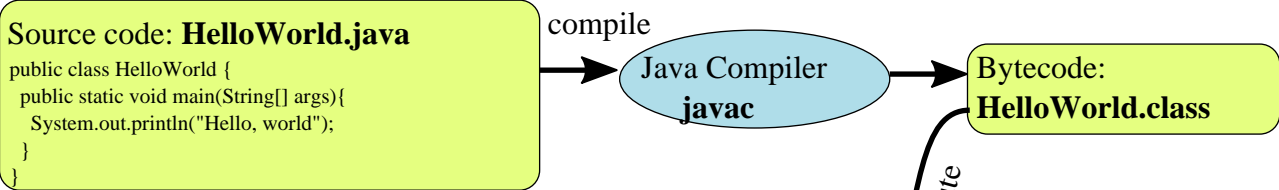




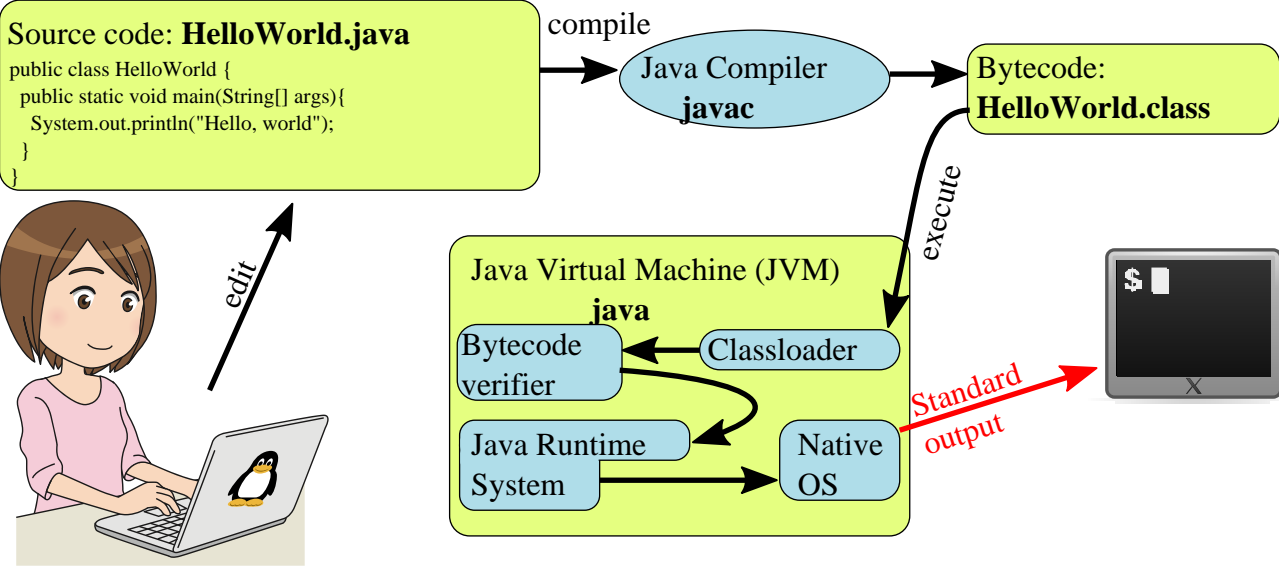
# Edit - compile - execute



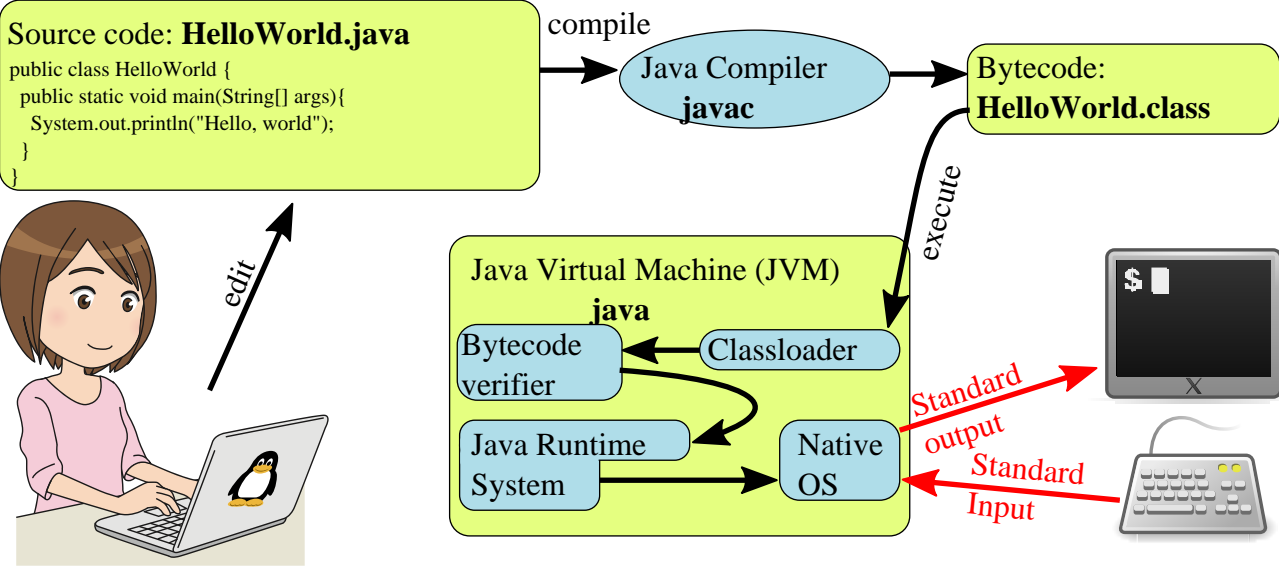
# Edit - compile - execute



# Edit - compile - execute



# Edit - compile - execute

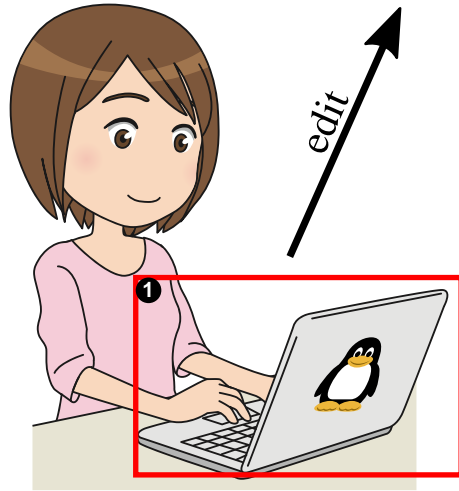


# Editing Java™ files

---

## 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 {
```

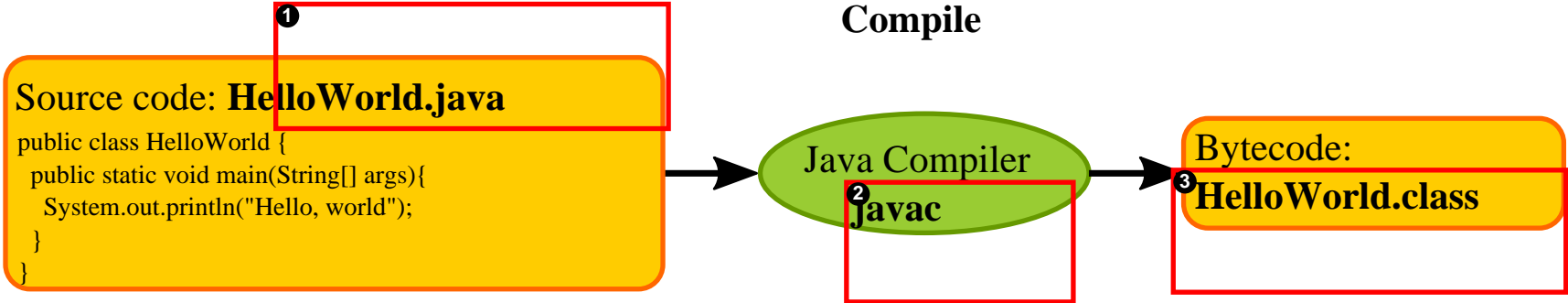
```
        System.out.println("Hello, world"); ❹
```

```
    }
```

```
}
```

# 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

```

Ëp°¾@@@6^@]
^@^F^@^O      ^@^P^@^Q^H^@^R
^@^S^@^T^G^@^U^G^@^V^A^@^F<i n i t >^A^@^C() V^A^@^DCode^A^@^CLi neNumberTabl e^A^@^Dnai n^A^@^V(\
  [ Lj ava/l ang/St r i n g; ) V^A^@
SourceFile^A^@^OHelloWorld.java^L^@^G^@^H^G^@^W^L^@^X^@^Y^A^@^LHello, world^G^@^Z^L^@^[ ^@
  ^\^A^@
HelloWorld^A^@^Pj ava/l ang/Obj ect ^A^@^Pj ava/l ang/Syst em^A^@^Cout ^A^@^ULj ava/i o/Pri nt Stream \
  ^A^@^Sj ava/i o/Pri nt Stream^A^@^Gp r i n t l n^A^@^U( Lj ava/l ang/St r i n g; ) V^@! ^@^E^@^F^@^@^@^@^@
  B^@^A^@^G^@^H^@^A^@      ^@^@^@] ^@^A^@^A^@^@^@^E*. ^@^A±^@^@^@^A^@
^@^@^@F^@^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

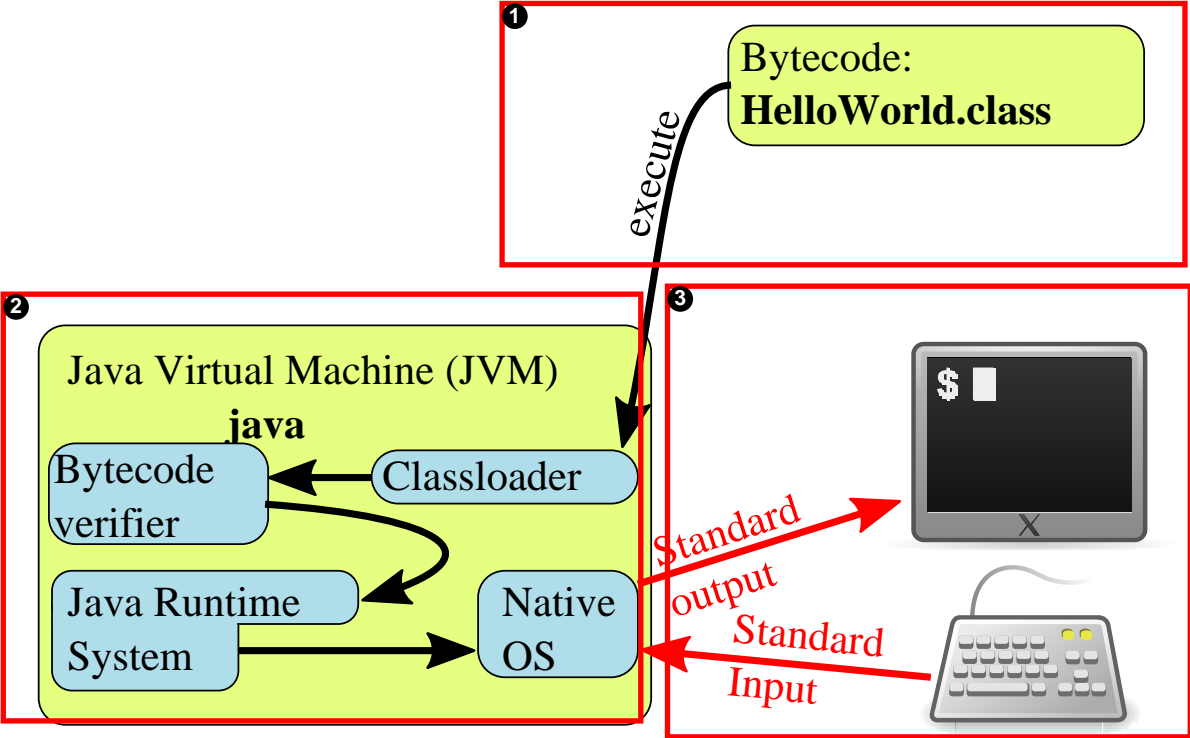
HelloWorld.java

- Human readable (kind of #).
- High abstraction level.
- Text file

HelloWorld.class

- 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.

# IntelliJ IDEA requires a JDK™

---

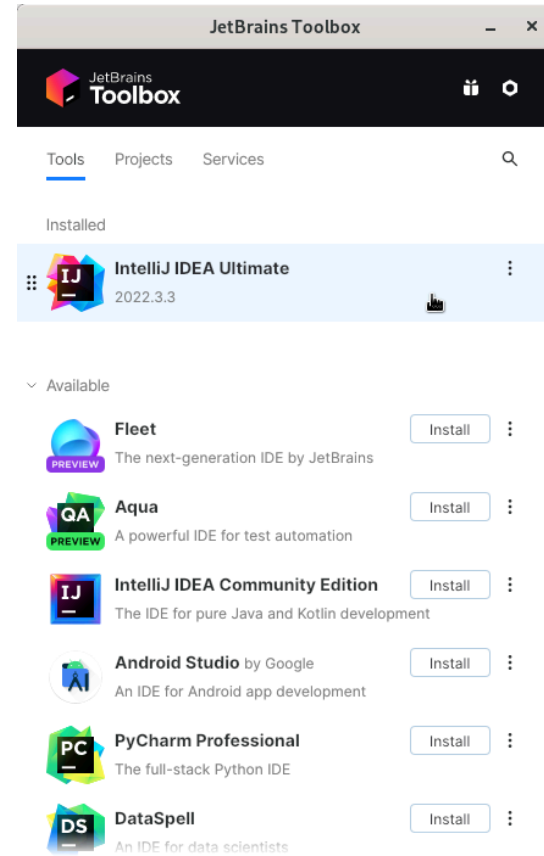
Prefer a **L**ong **T**erm **R**elease (**LTS**) e.g. 17, 21, ...

- Windows / Mac-OS: Manual Oracle or OpenJDK installation.
- Linux: Package install

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

# 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](http://www.jetbrains.com/shop/eform/students)

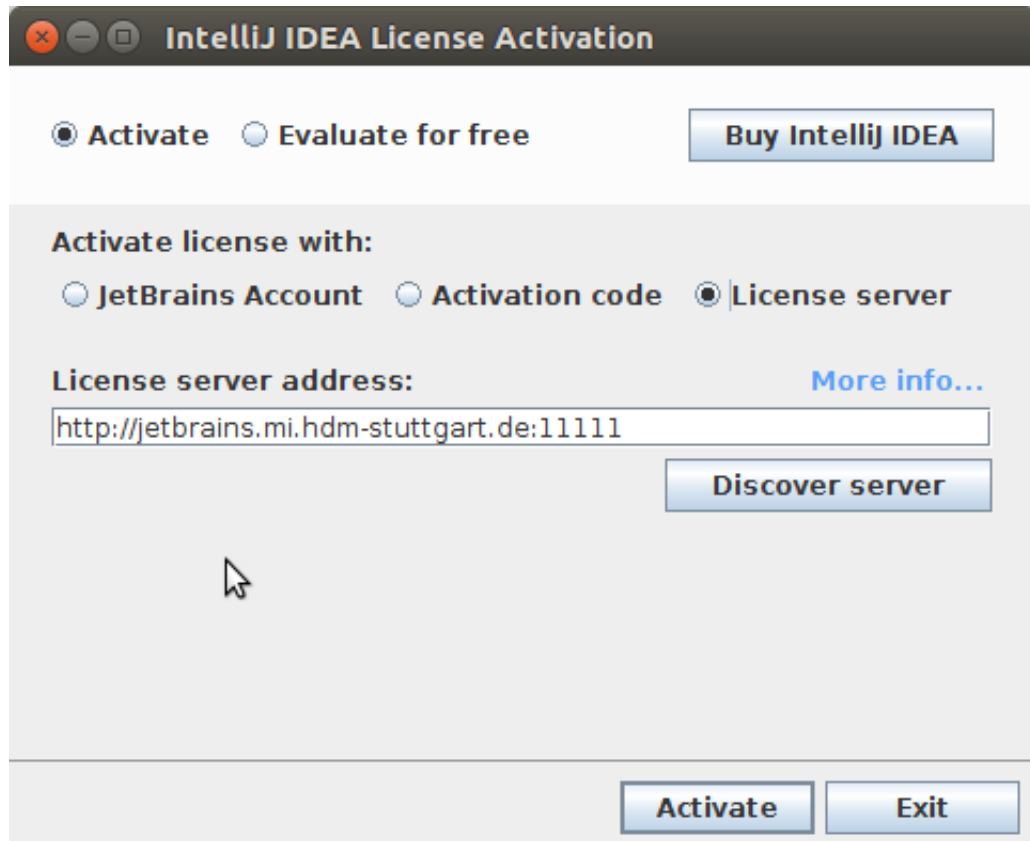
- Using HdM/MI license server:

**ht tp:** //jetbrains.mi.hdmstuttgart.de: 11111

See MI wiki for further details.

## Alternative: Using the HdM license server

---



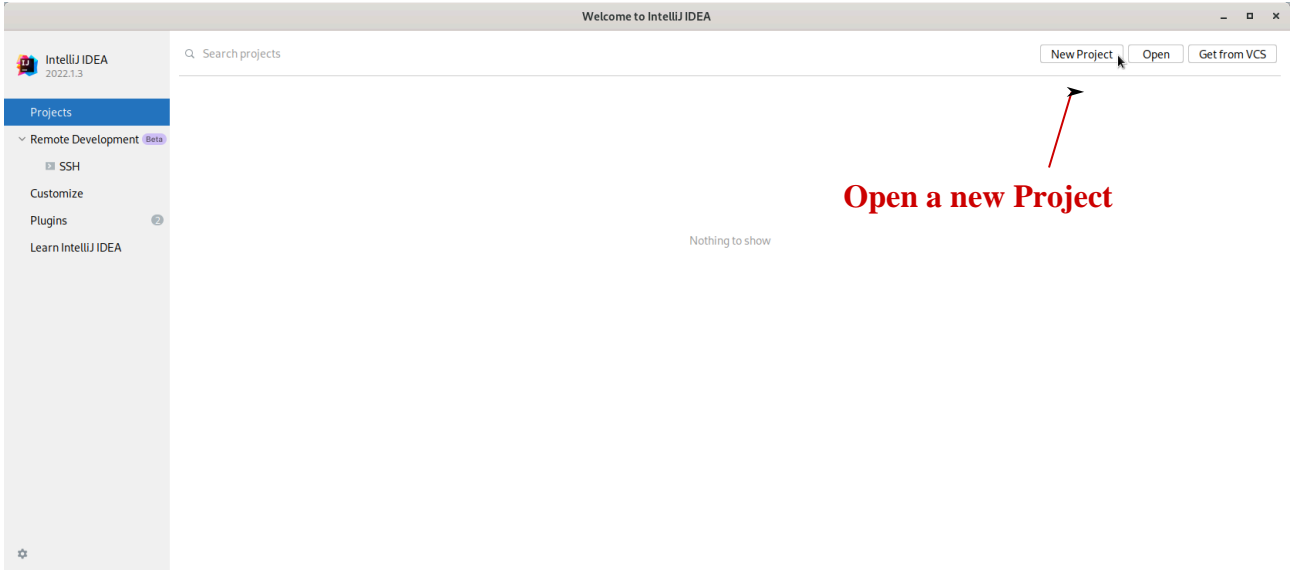
Insert address:

ht t p: //j et br ai ns. mi . hdm st ut t gart . de: 11111

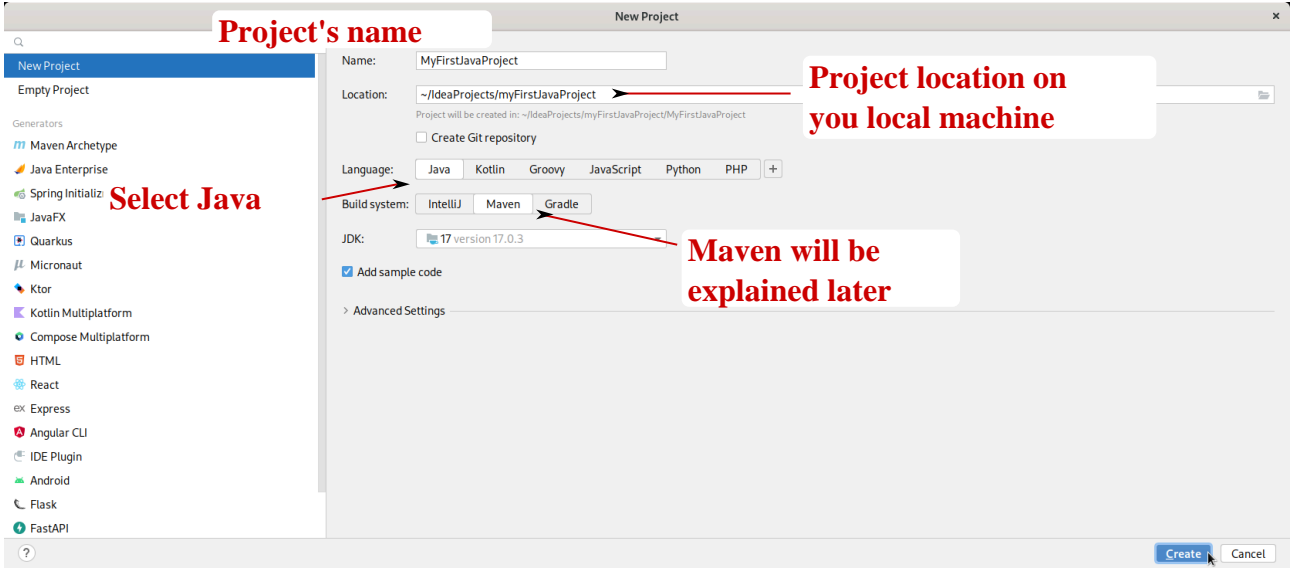
External usage requires VPN !



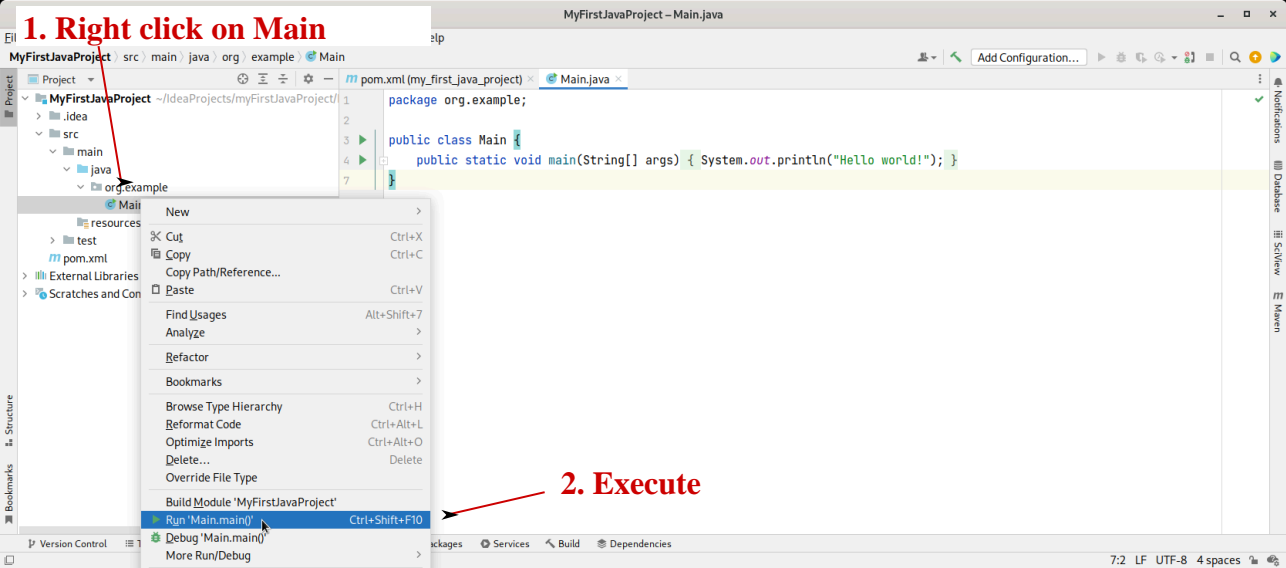
# Creating a new Java project



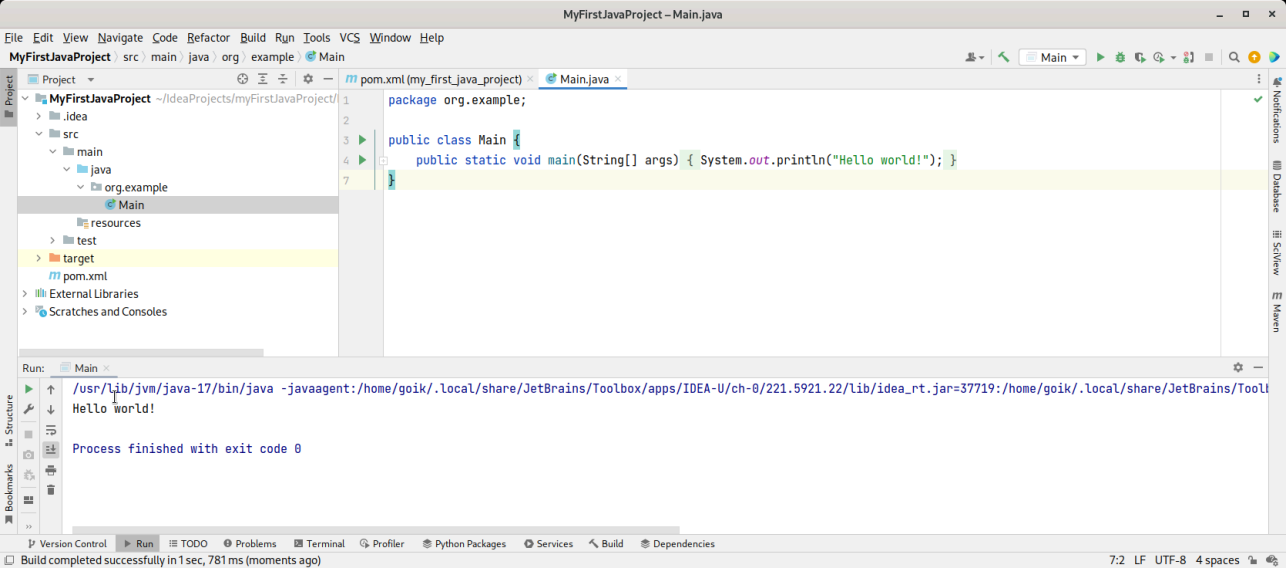
# Creating a new Java project



# Creating a new Java project



# Creating a new Java project



# 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! #)

## Related exercises

---

Exercise 1: Extending class HelloWorld

Exercise 2: Working with variables

Exercise 3: A conditional

Exercise 4: A loop