

# Overview

## Appendix

⇒ Examination bonus point projects



Is this your **TEAM**?

**T**oll

**E**in

**A**nderer

**M**achts

# Project rules

1. You are expected to work as a **team of three partners**.
2. Using the **MI Gitlab SCM** is a plus with respect to project evaluation. See table below.
3. Your team is expected to supply a Maven project based on the MI “Maven archetype quickstart” available from <https://maven.mi.hdm-stuttgart.de/nexus/repository/mi-maven/archetype-catalog.xml>.
4. You are expected to provide good internal code documentation with respect both to method signatures (**Javadoc™**) and method implementation.

You are expected to provide good internal code documentation with respect both to method signatures ([Javadoc™](#)) and method implementation. Possible problems involve:

```
24     value ++;
25
26
27     System.out.println("Value:" + value);
28
29
30     value++;
31
32     int result = (int) value / 2;
33
34 }
35
36
```

Lightbulb icon next to line 27.

Tooltip for line 32: Casting 'value' to 'int' is redundant [more...](#) (Ctrl+F1)

## Compile time warnings

Activate most compiler warnings at Editor --> Inspections. This will show potential compile time problems like dead / unnecessary / unreachable code, unused variable values, shadowing conflicts and so on.

Your method's formal parameters, their type and a method's return type must match your documentation.

```
5
6 ▶ public class ShowReachedPoints {
7
8     /**
9      * Execution reveals the number of reached p
10     *
11     * @param args Unused
12     */
13     Cannot resolve symbol 'args' more... (Ctrl+F1)
14     ▶ public static void main(String[] arg) {
15         RunTests.exec(
16             exerciseTitle: "Aufgabe 1" , Test_Helpe
17
18     }
```

## (Automated) tests

You are expected to provide meaningful unit tests:

- Try to cover all your implementation code and not just isolated modules / methods.
- If methods allow for null values write suitable tests.
- Test special cases: If a method expects i.e. an array of strings it may be allowed having zero length.

Your resulting project should be easily installable and runnable.

- Maven is a good starting point with respect both to testing and cross platform (Unix / Windows / Apple) portability.
- Avoid dependencies to local file system resources like `c : \users \xyz \testdata .txt`.

## Tip

Test your application's deployability by installing it on an untouched target platform (possibly of a different hard/software architecture) and execute `mvn test` (provided you do have written meaningful unit tests).

# Marking criteria

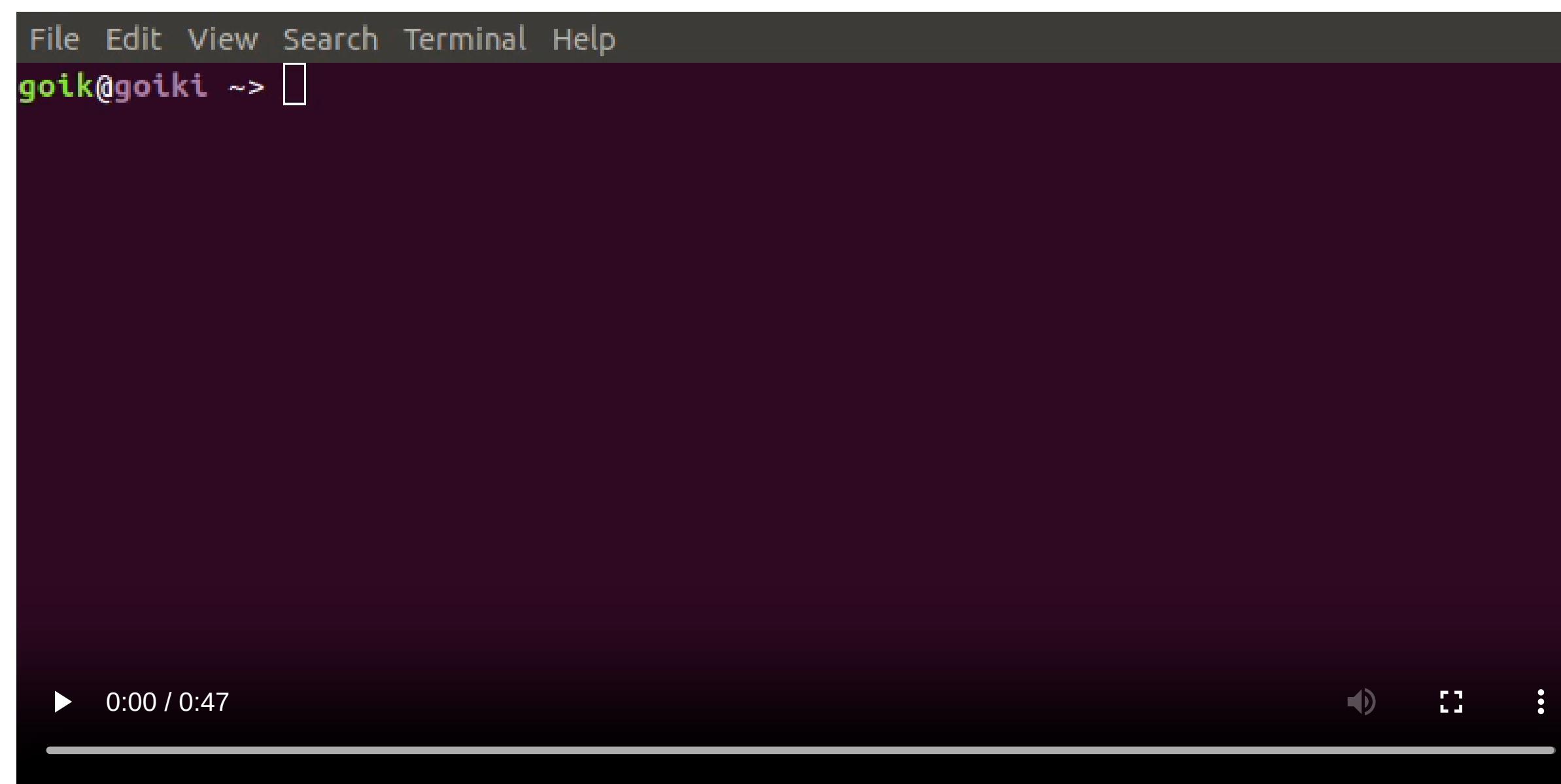
| Criterion              | Percentage |
|------------------------|------------|
| Overall code quality   | 20%        |
| Code documentation     | 20%        |
| Unit tests             | 10%        |
| Deployment             | 10%        |
| SCM usage              | 10%        |
| Software functionality | 30%        |

# Overview

## Appendix

- ⇒ Examination bonus point projects
  - ⇒ Weather forecast

# Sample forecast session



# Sample forecast invocation

```
goik@goiki target> java -jar weather-1.0.jar Stuttgart ①
1 = Stadtkreis Stuttgart ②
2 = Regierungsbezirk Stuttgart
3 = Stuttgart
4 = Stuttgart Feuerbach
5 = Stuttgart Muehlhausen

Bitte gültige Auswahl 1 bis 5 treffen: 2 ③
Vorhersage für Regierungsbezirk Stuttgart ④
Dienstag, 15.05
    23:00: 11°C, Leichter Regen
Mittwoch, 16.05
    02:00: 10°C, Leichter Regen
    05:00: 10°C, Leichter Regen
    08:00: 11°C, Leichter Regen
...
```

# Underlying data provider

```
https://api.openweathermap.org/data/2.5/forecast?lang=de&
APPID=7cufdhdcgdhsgdhgfcgsdss67b3&units=metric&id=3214105

{"cod":"200","message":0.0042,"cnt":40,"list":[
  {"dt":1526428800,"main":{"temp":10.29,"temp_min":10.29,
"temp_max":12.45,"pressure":985.75,"sea_level":1027.48,
"grnd_level":985.75,"humidity":80,"temp_kf":-2.16},
"weather":[{"id":500,"main":"Rain",
"description":"Leichter Regen","icon":"10n"}],"clouds":
{"all":88},"wind":{"speed":1.59,"deg":313.503},"rain":
{"3h":0.315},"sys":{"pod":"n"},"dt_txt":"2018-05-16 00:00:00"},
{"dt":1526439600,"main": ...
```

- 1 An **URL** containing an id value corresponding to a uniquely defined town or region. We identify the following components:

**lang=de**

Provide German localization e.g. «*Leichter Regen*» in favour of «light rain».

**APPID=7cufdhdcgdhsgdhgfcgsdss67b3**

This parameter allows for accessing the service: «7cufdhdcgdhsgdhgfcgsdss67b3» is actually a fake value. Your project requires **obtaining an APPID token**.

**units=metric**

`cities.list.json.gz` providing cities

```
[
  {
    "id": 2886241,
    "name": "Regierungsbezirk Köln",
    "country": "DE",
    "coord": {
      "lon": 7.16667,
      "lat": 50.833328
    }
  },
  {
    "id": 3247452,
    "name": "Kreis Euskirchen",
    ...
  ]
```

ma/Copy URL result to file

```
FileUtils.copyURLToFile(  
    "https://api.openweathermap.org/data/2.5/forecast...",  
    new File("weatherData.json"));
```

```
<dependency>  
  <groupId>commons-io</groupId>  
  <artifactId>commons-io</artifactId>  
  <version>2.6</version>  
</dependency>
```

# Parse city data

```
public class Cities {  
    static public final City[] cities;  
    ...  
}
```

```
@Test public void testParsedCityCount() {  
    Assert.assertEquals(209579, Cities.cities.length);  
}
```

# Parse weather data

```
public class WeatherDataParser {  
  
    static public final Weather parse(final String jsonWeatherDataFilename)  
        throws IOException {  
        return ...;  
    }  
}
```

```
@Test public void testParseWeatherData() {  
    ...  
    Weather weather = WeatherDataParser.parse(  
        "src/main/resources/stuttgart.weather.json");  
    ...  
}
```

# Requirements

1. The application shall accept a command line parameter like e.g. «Stuttgart» to filter matching cities from `cities.list.json`.
2. If a given filter matches multiple locations the user shall have an option for choosing the desired one.
3. The most recent city id value shall be cached in a file. Subsequent invocations without command line parameter shall provide a current forecast corresponding to this value.
4. Weather data belonging to a given id value shall be cached locally for 10 minutes. Subsequent weather queries within this period shall be read from cache rather than by accessing `https://api.openweathermap.org/ . . .`  
  
Provide logging to a file rather than to the console to avoid cluttering the user interface. Log cache handling.

# Logging

- Provide logging to a file rather than to the console to avoid cluttering the user interface.
- Log cache handling.

```
main INFO  weather.Forecast - Re-using cache file  
'/ma/goik/Forecast/6930414.json' from 196 seconds ago
```

## Appendix

- ⇒ Examination bonus point projects
  - ⇒ Reverse Polish notation (**RPN**) calculator
  - ⇒ Implementation hints

# Minimalist token scanner

```
final String[] patterns = new String[] {
    "sqrt",
    "[ -]?([0-9]+[.]?[0-9]*|.[0-9]+)(E[ -]?[0-9]+)?", // Matches e.g. -1.5E-33
    "\\+"}; // Escape required avoiding regular expression syntax clash.
final String expression = "2.1 -3.4 sqrt";
try (final Scanner scanner = new Scanner(expression)) {
    while (scanner.hasNext()) {
        for (final String p: patterns) {
            if (scanner.hasNext(p)) {
                System.out.println("Token '" + scanner.next(p) +
                    "' matched by '" + p + "'");
                break;
            }
        }
    }
}
```

Scanner output -1.34 0.34 + sqrt

Token '2.1' matched by '[-]?([0-9]+[.]?[0-9]\*|.[0-9]+)(E[-]?[0-9]+)?'  
Token '-3.4' matched by '[-]?([0-9]+[.]?[0-9]\*|.[0-9]+)(E[-]?[0-9]+)?'  
Token 'sqrt' matched by 'sqrt'

# Adding error detection

```
...
final String[] patterns = new String[] {
    "sqrt",
    "[-]?([0-9]+[.]?[0-9]*|.[0-9]+)(E[-]?[0-9]+)?", // Matches e.g. -1.5E-33
    "\\+"}; // Escape required avoiding regular expression syntax clash.
try (final Scanner scanner = new Scanner("2.1 -3.4 pbck") ❶) {
    while (scanner.hasNext()) {
        boolean foundToken = false;
        for (final String p: patterns) {
            if (scanner.hasNext(p)) {
                foundToken = true;
                System.out.println("Token '" + scanner.next(p) +
                    "' matched by '" + p + "'");
                break;
            }
        }
        if (!foundToken) {
            System.out.println("Parsing error at '" + scanner.nextLine() + "'");
            System.exit(1);
        }
    }
}
```

# Error message

Token '2.1' matched by '[-]?([0-9]+[.]?[0-9]\*|.[0-9]+)(E[-]?[0-9]+)?'

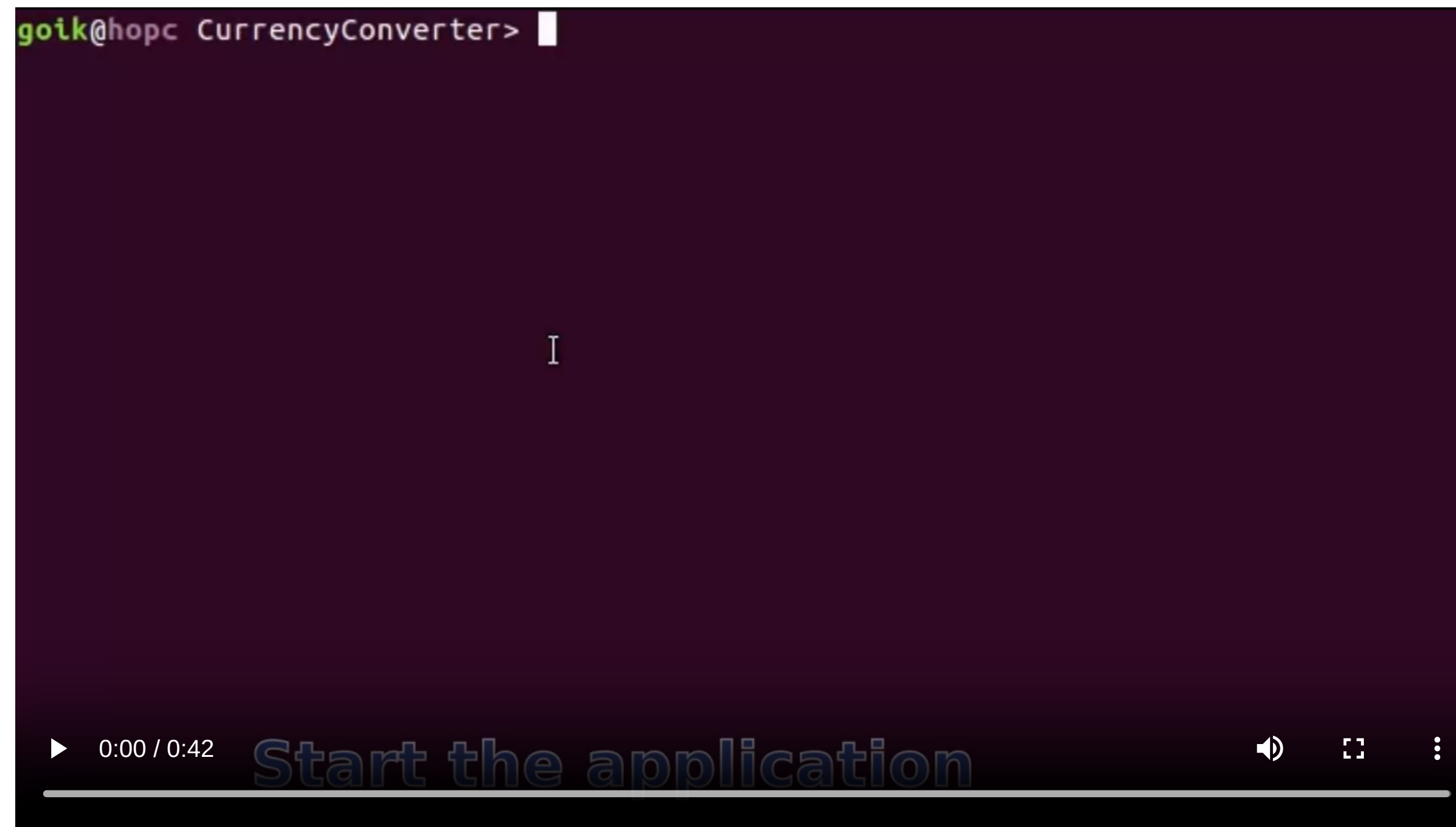
Token '-3.4' matched by '[-]?([0-9]+[.]?[0-9]\*|.[0-9]+)(E[-]?[0-9]+)?'

**Parsing error at 'pbck'**

## Appendix

- ⇒ Examination bonus point projects
  - ⇒ Currency converter, Summer 2017

Running the currency converter terminal application.



Overview

Appendix

⇒ Exercising past examinations

# Exam training by Guacamole

guacamole.mi.hdm-stuttgart.de/#/client/MTkyLjE2OC42LjE2OC8yMTIAYwBsZGFw

**bwLehrpool** vmChooser

Natives Linux | Zuletzt verwendet | Alle Kurse

▼ Kursumgebungen speziell für diesen Raum

- 113222a\_Datenbanken
- Mensch-Computer-Interaktion (HCI) 119203a Probeklausur
- COPT Oxford Online Placement Test
- Web Development

▼ Allgemeine Kursumgebungen

- 113219 Linux Essentials
- 113482a Software Engineering (Bachelor)
- E-Exam 113105 Softwareentwicklung 1 Test / Network blocked**
- E-Exam 113105 Softwareentwicklung 1 Test / Teamviewer**
- E-Exam Mathematik Test
- MI Linux Standard
- Programming Massively Parallel Processors

Liste filtern:

Fehler melden | Hilfe/News

**Details**

Name: E-Exam 113105 Softwareentwicklung 1 Test / Network blocked

Zuständig:

Betriebssystem: Ubuntu 16.4 64-bit

Plattform: VMware

Beschreibung:

Kann außerhalb von Prüfungszeiten gestartet werden. Zugriff auf externe Seiten durch lokale Firewall eingeschränkt.

☐ Bildschirmschoner deaktivieren

☐ IWS-Teilnahme

Abbrechen | Start

**»Teamviewer«: No network blocking**

## Environment hints:

The following installations allow for exercises with respect to the final examination:

- E-Exam 113105 *Softwareentwicklung 1* Test / Network blocked:

The examination environment with firewall rules restricting Internet access to a small number of allowed sites.

- E-Exam 113105 *Softwareentwicklung 1* Test / Teamviewer:

No Internet blocking. [Teamviewer](#) allows for getting external help.

## Appendix

- ⇒ Exercising past examinations
- ⇒ Starting an exam

# Preparing an examination

The screenshot shows a web browser window with the following elements:

- Browser Tab:** MI E-Exams - ILIAS Login P x
- Address Bar:** klausur.mi.hdm-stuttgart.de/ilias/login.php?target=&client\_id=iliasclient1&auth\_stat=
- Page Header:** MI Klausuren (left) and Language (dropdown) (right).
- Logo:** Hochschule der Medien (hdM) logo in the top left.
- Section Header:** LOGIN TO ILIAS
- Form Fields:**
  - Username \*** (required): mistudent
  - Password \*** (required): masked with dots
- Buttons:** A blue Login button with a mouse cursor icon.
- Footer:** Links for New Account, Registration, Forgot your password?, Forgot your username?, and Terms of Service.

**\* Required**

**Login using your hdM account**

## Appendix

- ⇒ Exercising past examinations
- ⇒ Implementing the project skeleton

# Generating Javadoc™.

- HTML is difficult to read:

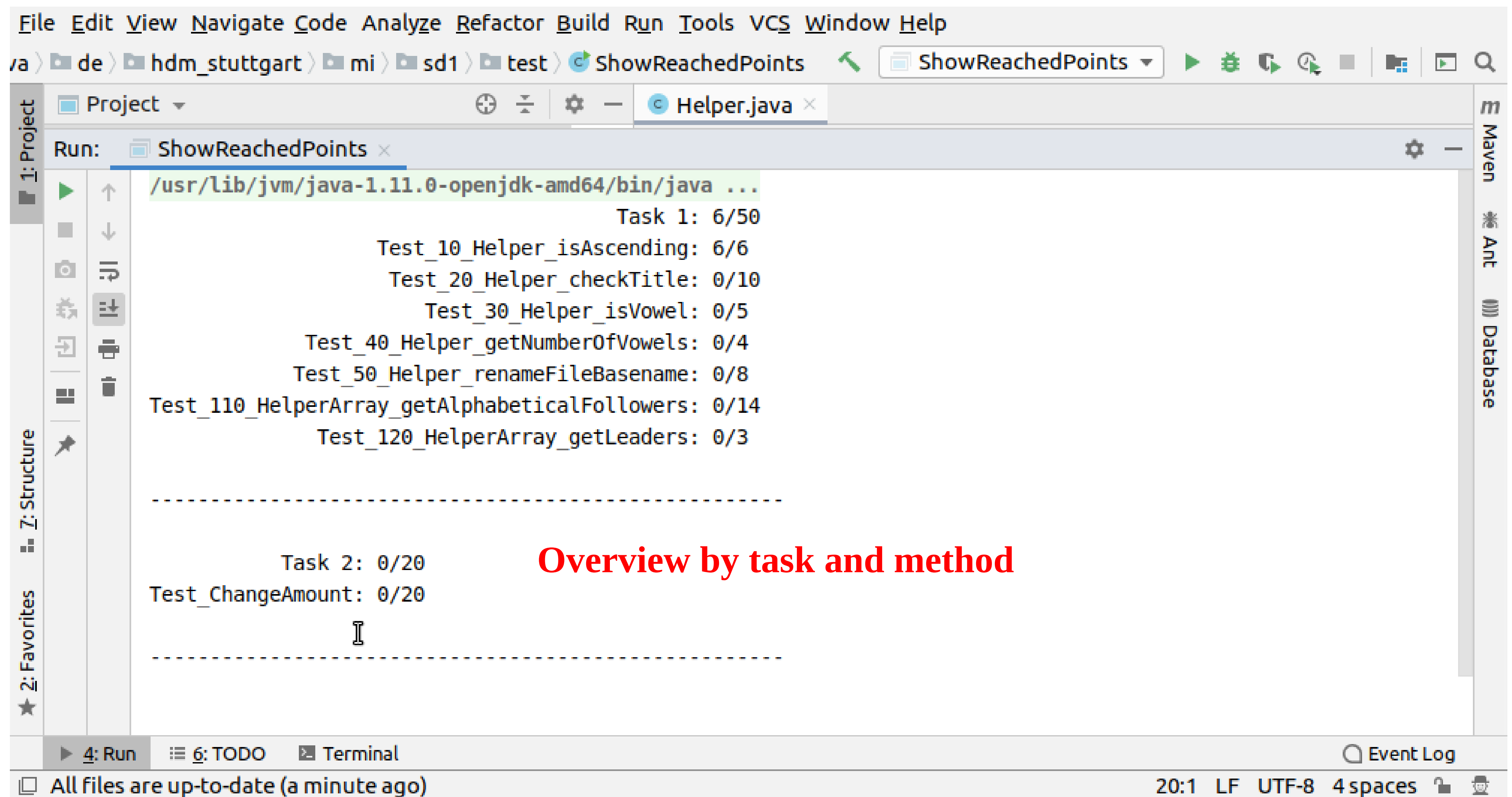
```
*      <td><code>a = -3</code>, <code>b = 4</code>, <code>c = 3</code>
*      <td>4 - (-3) = 7</td>
*    </tr>
*  </table>
*
*  ...
*/
static public int getMaxAbsoluteDiff(int a, int b, int c) {
    return 42; // TODO: Implement me correctly
```

- ⇒ Generate Javadoc™ by CLI or Idea.

# Programming hints

- The debugger is your friend ...
- ... but only if acquiring prior proficiency.
- Train **prime or related example**.

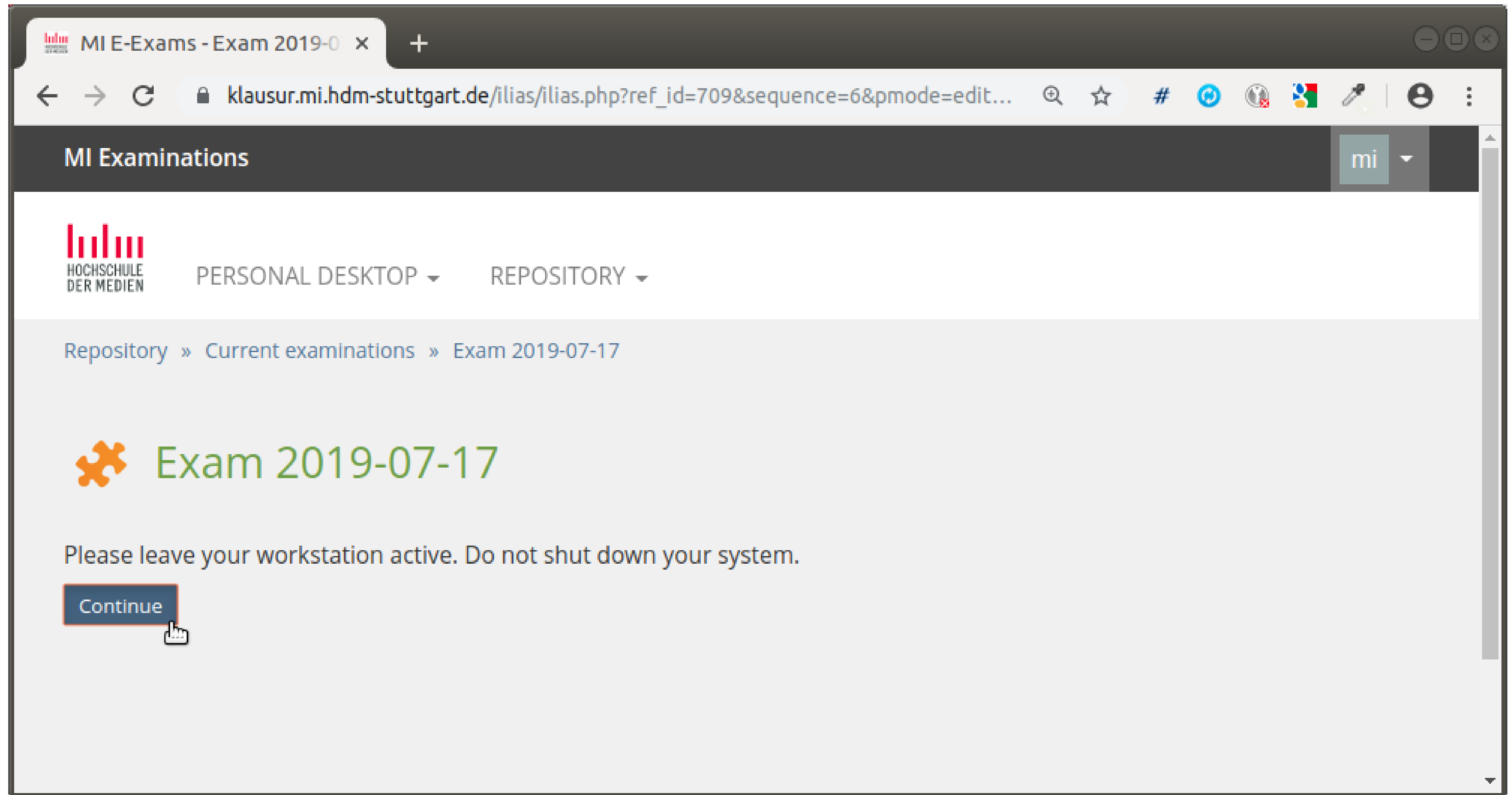
# The implement - test - implement cycle



## Appendix

- ⇒ Exercising past examinations
- ⇒ Finish the exam

# Finishing the exam



The screenshot shows a web browser window with a single tab titled "MI E-Exams - Exam 2019-07-17". The address bar displays the URL: `klausur.mi.hdm-stuttgart.de/ilias/ilias.php?ref_id=709&sequence=6&pmode=edit...`. The page header includes the text "MI Examinations" and a user profile icon labeled "mi". Below the header, the logo for "HOCHSCHULE DER MEDIEN" is visible, along with navigation links for "PERSONAL DESKTOP" and "REPOSITORY". The main content area shows a breadcrumb trail: "Repository » Current examinations » Exam 2019-07-17". Below this, there is an orange puzzle piece icon followed by the text "Exam 2019-07-17" in green. A message in blue text states: "Please leave your workstation active. Do not shut down your system." At the bottom left, there is a blue button labeled "Continue" with a mouse cursor hovering over it.

MI E-Exams - Exam 2019-07-17

klausur.mi.hdm-stuttgart.de/ilias/ilias.php?ref\_id=709&sequence=6&pmode=edit...

MI Examinations

mi

HOCHSCHULE  
DER MEDIEN

PERSONAL DESKTOP

REPOSITORY

Repository » Current examinations » Exam 2019-07-17

 Exam 2019-07-17

Please leave your workstation active. Do not shut down your system.

Continue

Overview

Appendix

⇒ Examination hints

# Personal examination cheat sheets

- Zip archive of **10 MB max**.

Caution: Use standard formats e.g. png, gif, pdf. Stuff like .doc, .docx, .rtf may be unsupported

- Upload your cheat sheet [here](#)

Caution: **Unavailable during active e-examinations!**

- Verify your uploaded cheat sheet at [here](#).

Caution: **Limited to HdM network / VPN**

# Unit tests in examinations

- Task definitions by **Javadoc™**.
- Corresponding Unit tests.
- Automated evaluation scoring your achievements.
- Individual weights reflecting a test's significance.

# Example interface definition

```
/**
 * Finde das n-te ungerade Element einer Wertefolge.
 *
 * <p>Beispiel: Im Array {3, 2, 0, 1, 4} ist der Wert «1» an der Index-
 * position «3» das zweite ungerade Element.</p>
 *
 * @param werte Die zu durchsuchenden Werte.
 * @param n Die gewünschte Position, Start bei 1.
 *
 * @return Den Index des n-ten ungeraden Wertes falls es mindestens n
 * ungerade Werte gibt, ...
 */
static public int getNtesUngeradesElement(final int[] werte, final int n){
    return 12345; // TODO: Implementiere mich korrekt!
}
```

# Corresponding test

```
@Test
@Marking(points=1) /* 1 point if test passes */
public void test_400() {
    Assert.assertEquals(
        2, /* Expected result */
        Helper.getNtesUngeradesElement(new int[]{-4, 6, 1, -2, 8}, 1));
}
```

Don't cheat!

| Unit Tests                                                                                                                                                                                                                                                            | Your solution                                                                                                                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>assertFalse(isPrime(1)); assertTrue (isPrime(2)); assertTrue (isPrime(3)); assertFalse(isPrime(4)); assertTrue (isPrime(5)); assertFalse(isPrime(6)); assertTrue (isPrime(7)); assertFalse(isPrime(8)); assertFalse(isPrime(9)); assertFalse(isPrime(10));</pre> | <pre>... boolean isPrime(final int p) {     switch (p) {         case 2:         case 3:         case 5:         case 7:             return true;         default:             return false;    }</pre> |

Will be treated as an **attempt at deception** / *Täuschungsversuch*.

# Unit tests strategy in examinations

- Unit testing is relentless: You are no longer at high school where a result having “just” a wrong sign used to matter next to nothing.
- Focus on completing units of work rather than “nearly” finishing a large number of tasks.
- Watching a test fail just happens. Learn to *systematically* fix bugs:
  1. Use your **IDE**'s debugger. Practise debugging **Junit** tests individually addressing failures one by one.
  2. Insert log statements using **log4j**.

## Followup exercises

210. Task

211. Swapping two variables

212. 0 and null

213. Task

214. Executing main(...)

215. Strange arithmetics

216. Task

217. Beginner's nightmare

218. Strange endless loop result

219. Task

220. switch versus if ... else if ... else

221. Exception handling

222. Task

223. Extraterrestrial integer value representation

224. An initialization issue

225. Tasks

226. equals ( ) and hashCode ( )

227. The average of three byte values

Overview

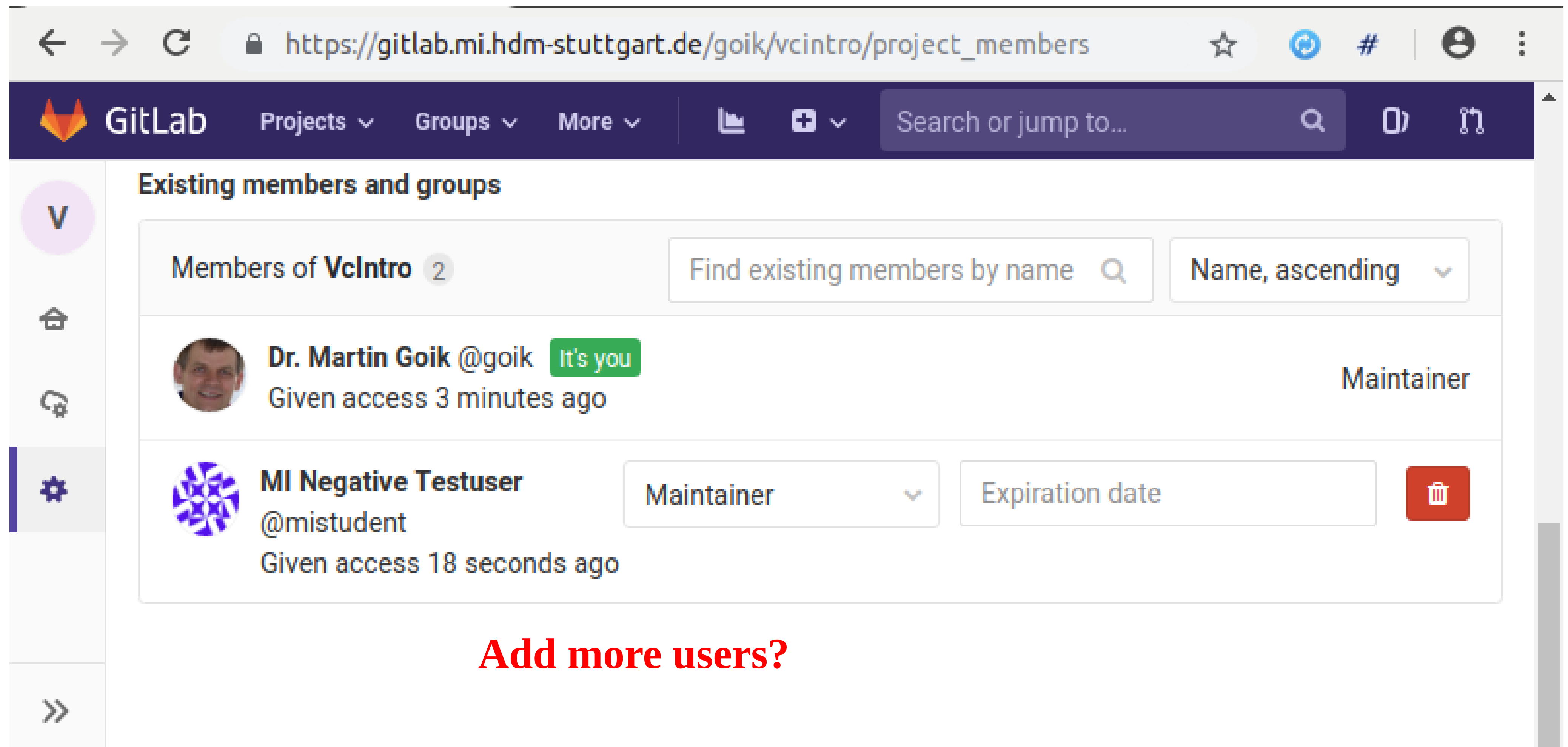
Appendix

→ Working with git

## Steps creating a new project

1. Creating an empty new project itself.
2. Adding fellow project users for participation.

# Creating a project at MI gitlab



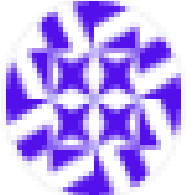
The screenshot shows the GitLab web interface for a project named 'vcintro'. The browser address bar displays the URL: `https://gitlab.mi.hdm-stuttgart.de/goik/vcintro/project_members`. The GitLab logo and navigation menu are visible at the top. The left sidebar contains icons for the user profile, home, search, and settings. The main content area is titled 'Existing members and groups' and shows a list of project members. The first member is 'Dr. Martin Goik @goik', who is the user 'It's you' and has been given access 3 minutes ago. The second member is 'MI Negative Testuser @mistudent', who has been given access 18 seconds ago. The interface includes search and sorting options for the members list.

Existing members and groups

Members of **VcIntro** 2

Find existing members by name

Name, ascending

|                                                                                     |                                                                                  |            |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|------------|
|  | <b>Dr. Martin Goik</b> @goik <span>It's you</span><br>Given access 3 minutes ago | Maintainer |
|  | <b>MI Negative Testuser</b> @mistudent<br>Given access 18 seconds ago            | Maintainer |

Expiration date

**Add more users?**

# Cloning a git project

```
>git clone git@gitlab.mi.hdm-stuttgart.de:goik/vcintro.git  
Cloning into 'vcintro'...  
warning: You appear to have cloned an empty repository.
```

Enter project folder, add **Readme .md**

```
>cd vcintro/  
>vim Readme.md  
>git add Readme.md
```

```
# Initial project description.  
  
Will be extended when adding more assets.
```

# Committing change set

```
EDITOR=vim git commit Readme.md
```

---

## **Adding Readme file in Markdown format**

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file:   Readme.md
```

# Push to upstream repository

```
>git push
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 306 bytes | 306.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To gitlab.mi.hdm-stuttgart.de:goik/vcintro.git
* [new branch]      master -> master
```

# Inserting a Maven project

```
>mvn --batch-mode -e archetype:generate -Dversion=0.9 \  
> -DgroupId=de.hdm_stuttgart.mi.sd1 \  
> -DartifactId=first -DarchetypeGroupId=de.hdm_stuttgart.mi \  
> -DarchetypeArtifactId=mi-maven-archetype-quickstart -DarchetypeVersion=1.2.1
```

```
>find first/ -type f  
first/.gitignore  
first/src/test/java/de/hdm_stuttgart/mi/sd1/AppTest.java  
first/src/main/java/de/hdm_stuttgart/mi/sd1/App.java  
first/src/main/resources/log4j2.xml  
first/pom.xml
```

# git status 1

```
> git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    first/

nothing added to commit but untracked files present (use "git add" to track)
```

# Adding Maven files to repository

```
>git add `find first/ -type f`  
>git status  
On branch master  
Your branch is up to date with 'origin/master'.  
  
Changes to be committed:  
  (use "git reset HEAD <file>..." to unstage)  
  
    new file:   first/.gitignore  
    new file:   first/pom.xml  
    new file:   first/src/main/java/de/hdm_stuttgart/mi/sd1/App.java  
    new file:   first/src/main/resources/log4j2.xml  
    new file:   first/src/test/java/de/hdm_stuttgart/mi/sd1/AppTest.java
```

# git status 2

```
>git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   first/.gitignore
    new file:   first/pom.xml
    new file:   first/src/main/java/de/hdm_stuttgart/mi/sd1/App.java
    new file:   first/src/main/resources/log4j2.xml
    new file:   first/src/test/java/de/hdm_stuttgart/mi/sd1/AppTest.java
```

# Commit Maven project files

```
EDITOR=vim git commit -a
```

---

## **Adding a Maven project.**

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Your branch is up to date with 'origin/master'.
#
# Changes to be committed:
#       new file:   first/.gitignore
#       new file:   first/pom.xml
#       new file:   first/src/main/java/de/hdm_stuttgart/mi/sd1/App.java
#       new file:   first/src/main/resources/log4j2.xml
#       new file:   first/src/test/java/de/hdm_stuttgart/mi/sd1/AppTest.java
```

# git status 3

```
>git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

# Push to upstream again

```
>git push
Counting objects: 22, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (22/22), 3.31 KiB | 1.10 MiB/s, done.
Total 22 (delta 0), reused 0 (delta 0)
To gitlab.mi.hdm-stuttgart.de:goik/vcintro.git
  32da2ff..4e19142  master -> master
```

# Reverting changes

```
>git status
...
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Readme.md
```

```
git checkout -- Readme.md
```

```
>git status
Your branch is up to date with 'origin/master'.
nothing to commit, working tree clean
```

# Pull changes from upstream

```
>git pull
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
From gitlab.mi.hdm-stuttgart.de:goik/vcintro
   3751344..83bd7b9  master    -> origin/master
Updating 3751344..83bd7b9
Fast-forward
 README.md | 3 +--
 1 file changed, 1 insertion(+), 2 deletions(-)
```

Overview

Appendix

→ Apache Maven

# Maven: Recommended reading

- [\[Maven2010\]](#)

# What is Maven anyway?

- Build tool
- Project management tool
  - Create reports
  - Continuous integration support

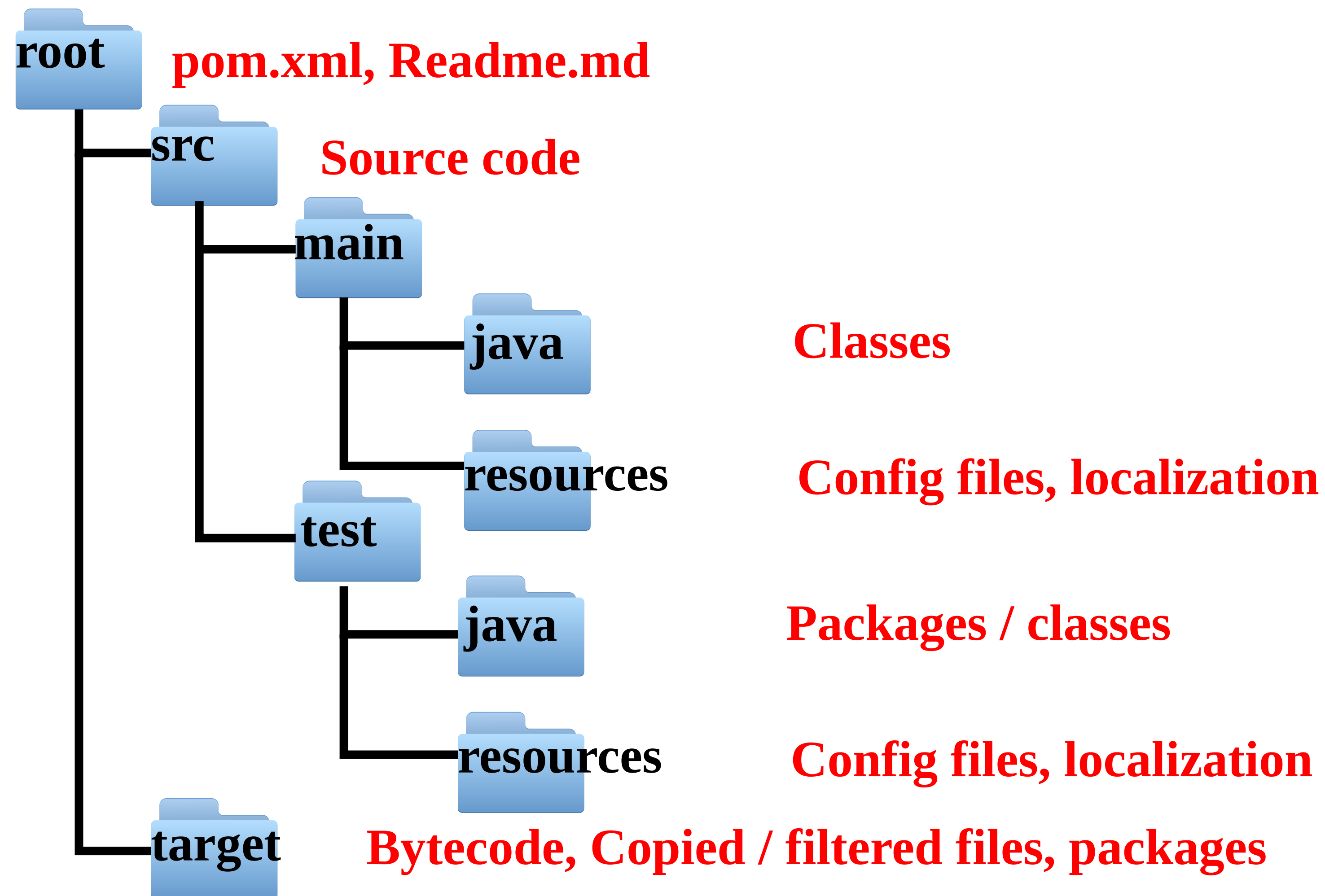
# Maven: Facts and benefits

- Build tool
- Dependency management
- Repository system
- Plugin framework

# Convention Over Configuration

- Sensible default values:
  - Source below `${basedir}/src/main/java`
  - Tests below `${basedir}/src/test`
  - Bytecode, jar/war archives below `${basedir}/target`
  - ...

# Maven project layout



# The project object model file

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  <a class="uri" href="http://maven.apache.org/xsd/maven-4.0.0.xsd" target="_blank">http://maven.apache.org/xsd/mav
"> <modelVersion>4.0.0</modelVersion> ... </project>
```

# pom.xml characteristics

- Declarative project description
  - Dependencies
  - Builds
  - Artifacts
- No explicit instructions

# pom.xml vs. Makefile

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-shade-plugin</artifactId>
  <configuration>
    <Main-Class>org.devel.App</Main-Class>
  ...
    <execution>
      <phase>package</phase>
      <goals>
        <goal>shade</goal>
      </goals>
  ...
```

```
module.o: module.c
gcc -c -g module.c
```

# Overview

## Appendix

- ⇒ Apache Maven

- ⇒ The project object model pom.xml

“Hello, world” pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>
  <groupId>de.hdm_stuttgart.mi</groupId>
  <artifactId>first</artifactId>
  <version>0.9</version>

</project>
```

# Executing “compile” phase

```
mkdir -p src/main/java ❶
```

```
vim src/main/java/Hello.java ❷
```

```
first> mvn compile ...
```

```
[WARNING] File encoding has not been set, using platform encoding UTF-8, ❸  
         i.e. build is platform dependent! ...
```

```
[ERROR] error: Source option 5 is no longer supported. ❹ Use 6 or later.
```

```
[ERROR] error: Target option 1.5 is no longer supported. ❺ Use 1.6 or later.
```

# Examining the Java™ version culprit

```
>mvn help:effective-pom
```

```
<project ...>
```

```
...
```

```
<plugin>
```

```
<artifactId>maven-compiler-plugin</artifactId> ...
```

```
> find ~/.m2/repository/ -name maven-compiler-plugin\* ...
```

```
>jar -xf ~/.m2/repository/org/apache/maven/plugins/maven-compiler-plugin/3.7.0/maven-compiler-plugin-3.7.0.jar
```

```
>cat META-INF/maven/plugin.xml
```

```
<encoding implementation="java.lang.String" default-value="${project.build.sourceEncoding
```

```
<source implementation="java.lang.String" default-value="1.5">${maven.compiler.source}</source>
```

```
<target implementation="java.lang.String" default-value="1.5">${maven.compiler.target}</target>
```

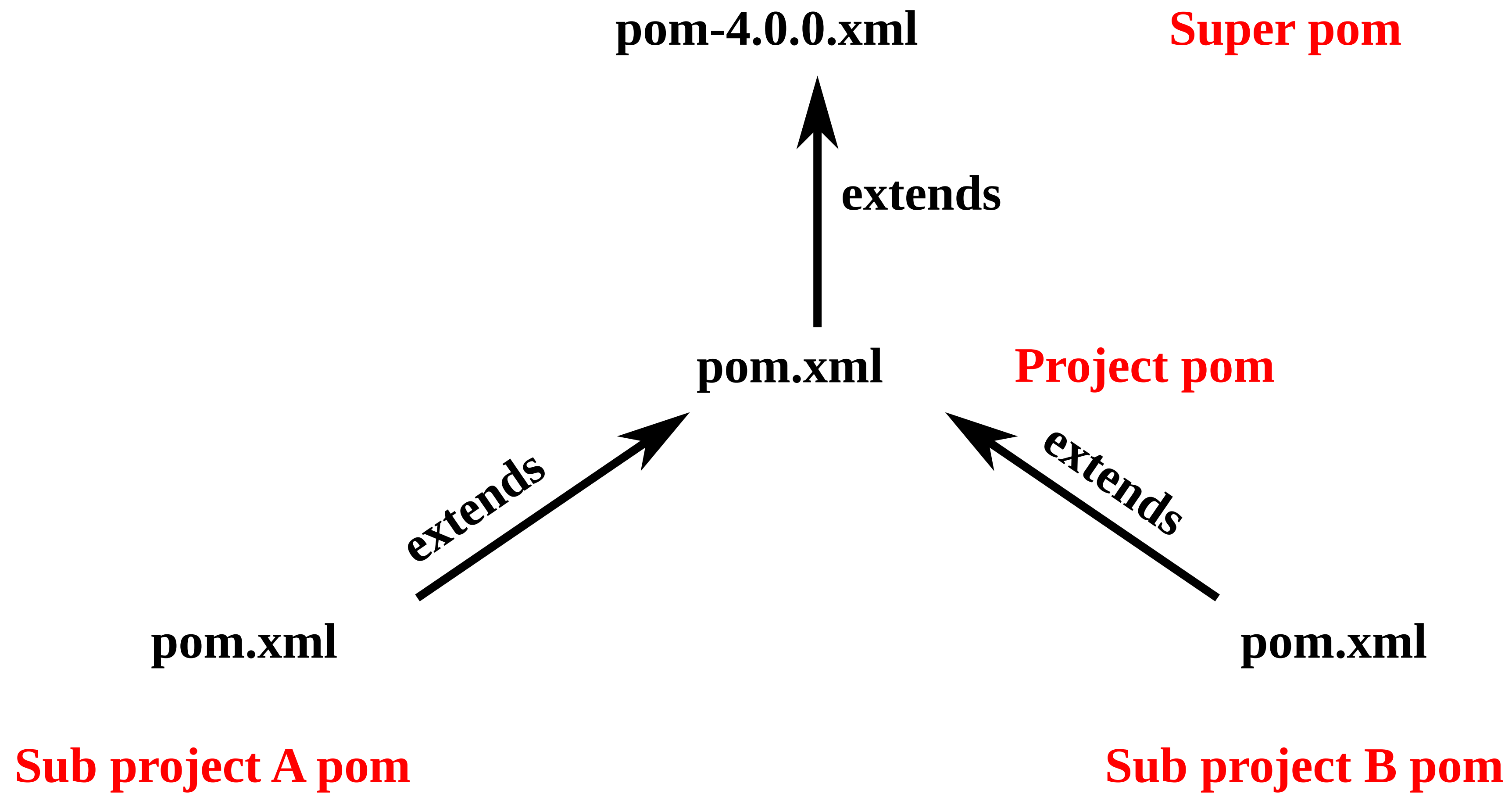
# Resolving encoding / Java™ version issues

```
<project ... xsd/maven-4.0.0.xsd">
...
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties> ...
```

```
rm -rf ~/.m2/repository/
mvn compile;      find ~/.m2/repository/ -type f|wc -l
220
```

# POM inheritance



# The Super POM

```
jar -tf /usr/share/apache-maven-3.0.5/lib/maven-model-builder-3.0.5.jar
```

```
...
```

```
org/apache/maven/model/plugin/ReportingConverter.class
```

```
org/apache/maven/model/pom-4.0.0.xml
```

```
org/apache/maven/model/profile/activation/FileProfileActivator$1.class
```

```
...
```

# pom-4.0.0.xml content

```
...                                <!-- Does this ring a (security related?) bell? -->
<repositories>
  <repository>
    <id>central</id>
    <name>Central Repository</name>
    <url>http://repo.maven.apache.org/maven2</url>
  ...
  <pluginRepositories>
    <pluginRepository>
      <id>central</id>
      <name>Central Repository</name>
      <url>http://repo.maven.apache.org/maven2</url>
    ...
  ...
```

Favour https in ~/ .m2/settings.xml

```
<settings ... settings-1.0.0.xsd">

  <mirrors>
    <mirror>
      <id>central-secure</id>
      <mirrorOf>central</mirrorOf>
      <name>Maven Central: Favour https over http.</name>
      <url>https://repo.maven.apache.org/maven2</url>
    </mirror>
  </mirrors>
  ...

```

# Resolving to effective pom.xml

```
first> mvn help:effective-pom
... Effective POMs, after inheritance, interpolation, and profiles are applied:

<project xmlns="http://maven.apache.org/POM/4.0.0" ...>
  <modelVersion>4.0.0</modelVersion>
  ...
    <plugin>
      <artifactId>maven-jar-plugin</artifactId>
      <version>2.3.2</version>
      <executions>
        <execution>
          <id>default-jar</id>
        </execution>
      </executions>
    </plugin>
  ...
</project>
```

Overview

Appendix

⇒ Apache Maven

⇒ Plugins

# Plugin architecture

- Tiny core
- Plugin extensible

Hint: `mvn help:effective-pom` lists included plugins.

## Sample plugins

- Testing: `maven-surefire-plugin`
- Library dependencies: `maven-dependency-plugin`
- Packaging: `maven-jar-plugin`
- Documentation: `maven-javadoc-plugin`

# Example: The maven-javadoc-plugin

```
first> jar -tf ~/.m2/repository/org/apache/maven/plugins/\  
      maven-javadoc-plugin/3.0.0/maven-javadoc-plugin-3.0.0.jar  
  
...  
META-INF/LICENSE  
META-INF/maven/org.apache.maven.plugins/maven-javadoc-plugin/plugin-help.xml  
javadoc-report.properties  
META-INF/maven/plugin.xml  
log4j.properties  
META-INF/NOTICE  
org/apache/maven/plugins/javadoc/AbstractFixJavadocMojo$JavaEntityTags.class  
org/apache/maven/plugins/javadoc/AggregorTestJavadocReport.class  
...
```

# Overview

## Appendix

- ⇒ Apache Maven

- ⇒ Dependencies

# Adding test capabilities

```
<project ... maven-4.0.0.xsd"> ...  
  <dependencies>  
    <dependency>  
      <groupId>junit</groupId>  
      <artifactId>junit</artifactId>  
      <version>4.12</version>  
      <scope>test</scope>  
    </dependency> ...
```

```
import org.junit.Test;  
import org.junit.Assert;  
  
public class AppTest {  
    @Test  
    public void doTest() {  
        Assert.assertEquals(1, 1);  
    }  
}
```

---

```
first> mvn test  
[INFO] Scanning for projects...  
Running de.hdm_stuttgart.mi.sd1.AppTest  
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.053 sec
```

# Dependency listing

```
first> mvn dependency:tree
...
[INFO] de.hdm_stuttgart.mi.sd1:first:jar:0.9
[INFO] \- junit:junit:jar:4.12:test
[INFO]    \- org.hamcrest:hamcrest-core:jar:1.3:test
```

- ~/ .m2/repository/**junit/junit/4.12/junit-4.12.jar**
- ~/ .m2/repository/**org/hamcrest/hamcrest-core/1.3/hamcrest-core-1.3.jar**

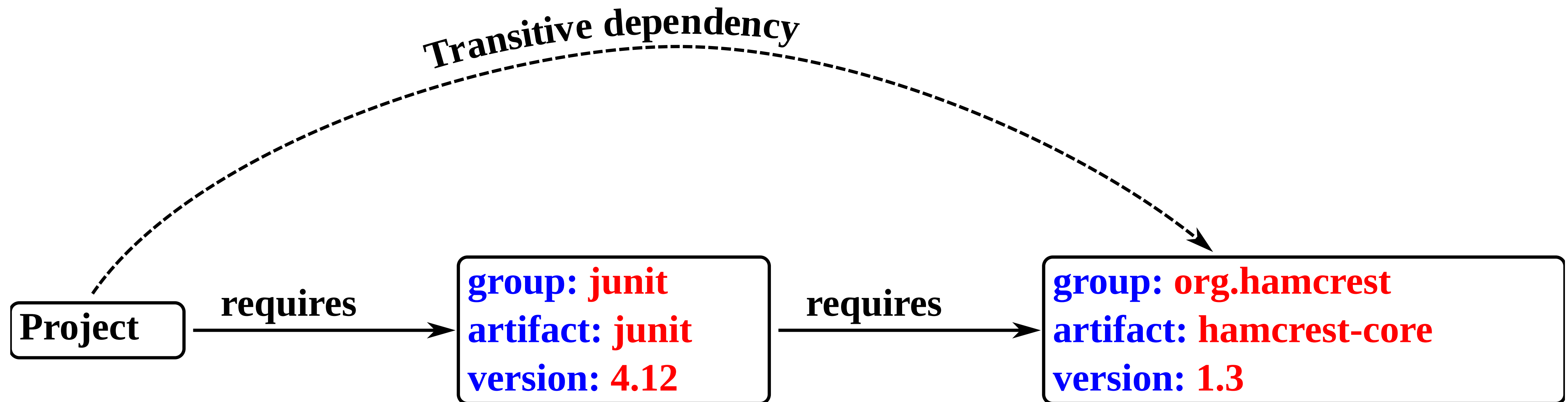
# Absence of hamcrest in pom.xml

```
<project ... maven-4.0.0.xsd">
  ...
  <!-- no such entry -->
  <dependency>
    <groupId>org.hamcrest</groupId>
    <artifactId>hamcrest-core</artifactId>
    <version>1.3</version>
    <scope>test</scope>
  </dependency>
  ...
```

~/ .m2/repository/junit/junit/4.12/junit-4.12.pom

```
<dependency>  
  <groupId>org.hamcrest</groupId>  
  <artifactId>hamcrest-core</artifactId>  
  <version>1.3</version>  
</dependency>  
</dependencies>
```

# Transitive dependencies



```
<dependency>
  <groupId>junit<...
  <artifactId>junit<...
  <version>4.12<...
</dependency>
```

```
<dependency>
  <groupId>org.hamcrest<...
  <artifactId>hamcrest-core<...
  <version>1.3<...
</dependency>
```

# Oblivious to test implementation: TestNG

```
<project ... maven-4.0.0.xsd"> ...  
  <dependencies>  
    <dependency>  
      <groupId>org.testng</groupId>  
      <artifactId>testng</artifactId>  
      <version>6.14.3</version>  
      <scope>test</scope>  
    </dependency> ...
```

```
import org.testng.annotations.Test;  
import org.testng.Assert;  
  
public class AppTest {  
  @Test  
  public void doTest() {  
    Assert.assertEquals(1, 1);  
  }  
}
```

---

```
testng> mvn test  
Running de.hdm_stuttgart.mi.sd1.AppTest  
Configuring TestNG with: org.apache.maven.surefire...  
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.372 sec
```

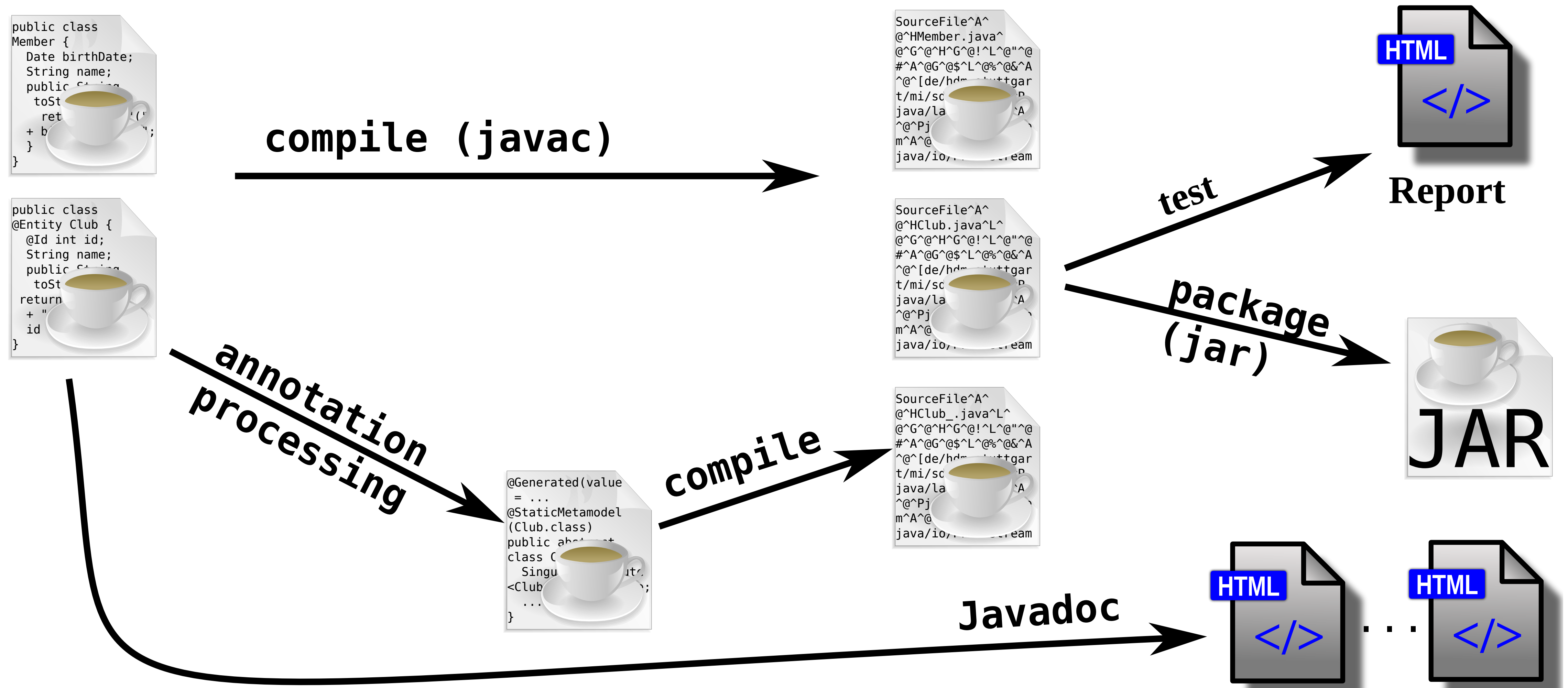
# Overview

## Appendix

- ⇒ Apache Maven

- ⇒ Lifecycle, phases and goals

# Phases



# Maven lifecycles and phases

## Phases:

clean, compile, test, package, deploy, ...

## Lifecycle

Sequence of [named phases](#)

## Example: mvn **clean** (Lifecycle)

- pre-clean
- clean (Phase)
- post-clean

See [Default Lifecycle](#).

# hooking into phase

```
<plugin>
  <groupId>com.mysema.maven</groupId>
  <artifactId>apt-maven-plugin</artifactId>
  <version>1.1.3</version>
  <executions>
    <execution>
      <id>process</id>
      <goals>
        <goal>process</goal>
      </goals>
      <phase>generate-sources</phase>
      <configuration>
        <outputDirectory>${project.build.directory}/metamodel</outputDirectory>
        <processor>com.mysema.query.apt.jpa.JPAAnnotationProcessor</processor>
      </configuration>
      ...
    </execution>
  </executions>
</plugin>
```