

# 1. Tutorium

## Orga, Java, Klassen und Objekte

### Tutorium 14

Péter Bohner | 02.11.2022



# Inhaltsverzeichnis

1. Organisatorisches
2. Artemis
3. Einrichtung Entwicklungsumgebung
4. Erstes Java Programm
5. Typen und Variablen
6. Klassen und Objekte
  - 6.1 Einführung
  - 6.2 Objekte
  - 6.3 Modellierung

Organisatorisches  
oooooooo

Artemis  
oooooooooooo

Einrichtung Entwicklungsumgebung  
oooo

Erstes Java Programm  
oooo

Typen und Variablen  
ooooo

Klassen und Objekte  
o  
o  
ooooo

# Vorstellung

## Über mich

- Péter Böhner
- Informatik 3. Semester
- `peter.bohner@student.kit.edu`

## Über euch

- Name
- Studiengang
- Programmiererfahrung
- Aktuelles/Liebblingsprojekt
- Erwartungen an dieses Tutorium

Organisatorisches  
●○○○○○○○○○

Artemis  
○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

# Das Tutorium

## Organisatorisches

- Mittwoch 15:45-17:15, 50.34, SR -107
- Folien im ILIAS (Tutorium 14)
- Für Fragen:
  - ILIAS-Forum
  - MS Teams
  - E-Mail: `peter.bohner@student.kit.edu`

## Ablauf

- Wiederholung des Vorlesungsstoffes mit Beispielen
- Übungsblattbesprechung
- Fragen beantworten

Organisatorisches  
○●○○○○○○○○○

Artemis  
○○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

## Ablauf

- Orientierungsprüfung:
  - Bis zum Ende des 2. Semesters versucht
  - Bis zum Ende des 3. Semesters bestanden
- Bestehen durch:
  - Übungsschein
  - 2 Abschlussaufgaben

Organisatorisches  
○○●○○○○○○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

## Übungsblätter

- 5 Blätter
- Bestanden: >50% der Übungspunkte
- Tipp: Die Blätter werden immer umfangreicher, holt eure Punkte am Besten am Anfang

## Präsenzübung

- Normale Prüfung (*vermutlich am 13.01.2022*)
- Überprüfung eures Grundwissens
- >75% der Punkte

Organisatorisches  
○○●○○○○○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

## Wichtig!

- **Übungsschein Voraussetzung für Abschlussaufgaben**
- Einzelabgaben werden automatisch auf Plagiate geprüft
- 1. Versuch: Ausschluss vom Übungsschein
- 2. Versuch: Exmatrikulation

Organisatorisches  
○○○●○○○○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

## Allgemein

- 20 Punkte pro Blatt
- alle zwei Wochen über ILIAS
- Abgabe nach spätestens 2 Wochen (donnerstags, 06:00 Uhr) über Artemis
- erstes Übungsblatt am 2. November (**heute!**)

## Bewertungskriterien

- Funktionalität: *Wird die Aufgabenstellung erfüllt?*
- Saubere Modellierung, Programmiermethodik
- Dokumentation (einheitlich Deutsch oder Englisch)

Organisatorisches  
○○○○●○○○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○



# Wichtige Fristen

## Fristen

- Anmeldung Artemis bis **02.11.2022, 12:00 Uhr** (also schon zu spät)
- Anmeldung Übungsschein mit Präsenzübung **07.12.2022, 12:00 Uhr**

Organisatorisches  
○○○○○●○○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

# Wichtige Links

## Alle Links und Materialien

- <https://bohner.me/tutorien/ws2223/>
- Ihr braucht also nicht mitzuschreiben

## Artemis

- <https://artemis.praktomat.cs.kit.edu/>
- Nur innerhalb des Uni-Netzes zugänglich

## VPN

- Ermöglicht, von außerhalb auf das Uni-Netz zuzugreifen
- <http://www.scc.kit.edu/dienste/openvpn.php> (OVPN)
- Mehr Infos im Wiki

Organisatorisches  
○○○○○○●○○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

# Wichtige Links

## ILIAS

- [Magazin » Organisationseinheiten » Fakultät für Informatik » WS 22/23 » Programmieren](#)
- [Programmieren-Wiki](#)
- [Tutoriumsordner » Tutorium 14](#)

Organisatorisches  
○○○○○○○●○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

## Was könnt ihr bei Fragen machen?

- 1 FAQ auf Vorlesungshomepage prüfen
- 2 Im Programmieren-Wiki im ILIAS nachschauen
- 3 Im Vorlesungsforum prüfen, ob die Frage bereits gestellt wurde
- 4 Im Vorlesungsforum nachfragen (aussagekräftige Titel verwenden)
- 5 Tutor fragen, im Tutorium oder per E-Mail
- 6 Email an [programmieren-vorlesung@cs.kit.edu](mailto:programmieren-vorlesung@cs.kit.edu)

Organisatorisches  
○○○○○○○○●○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

## Fragen?

Organisatorisches  
oooooooo●

Artemis  
oooooooooooo

Einrichtung Entwicklungsumgebung  
oooo

Erstes Java Programm  
oooo

Typen und Variablen  
ooooo

Klassen und Objekte  
o  
o  
ooooo

## Artemis

- neues Abgabesystem
- nutzt im Hintergrund GitLab und Jenkins
- Online-Editor: Code kann direkt komfortable in Artemis bearbeitet werden
- Oder: Aufgabe mit git clonen, lokal bearbeiten und Änderungen pushen

## Registrierung

- Registrierung drücken und mit KIT Account einloggen
- E-Mail mit Link zur Passwort-Setzung für Artemis (GitLab, Jenkins)
- Anmeldung mit Passwort und Kurs beitreten

Organisatorisches  
oooooooooooo

Artemis  
●ooooooooooooo

Einrichtung Entwicklungsumgebung  
ooooo

Erstes Java Programm  
ooooo

Typen und Variablen  
ooooo

Klassen und Objekte  
o  
o  
ooooo

## Git Basics

- `git clone url` - Clont das GitLab Repository lokal auf euren Rechner
- `git add file` - „staged“ (markiert) Datei für den nächsten Commit
- `git add -A` - Alle neuen/veränderten Dateien werden „gestaged“
- `git commit -m „Nachricht“` „gestagete“ Änderungen werden in einem Commit zusammengefasst
- `git pull` - Änderungen aus GitLab Repository werden heruntergeladen
- `git push` - Änderungen aus lokalen Repository werden hochgeladen
- mehr zu Git in SWT I oder in der [Git Dokumentation](#)
- ...oder Git über eclipse oder eine der vielen anderen [GUIs](#) benutzen

Nach jedem commit führt Artemis einen build aus und führt automatisiert die public Tests durch

Organisatorisches  
oooooooooooo

Artemis  
o●oooooooooooo

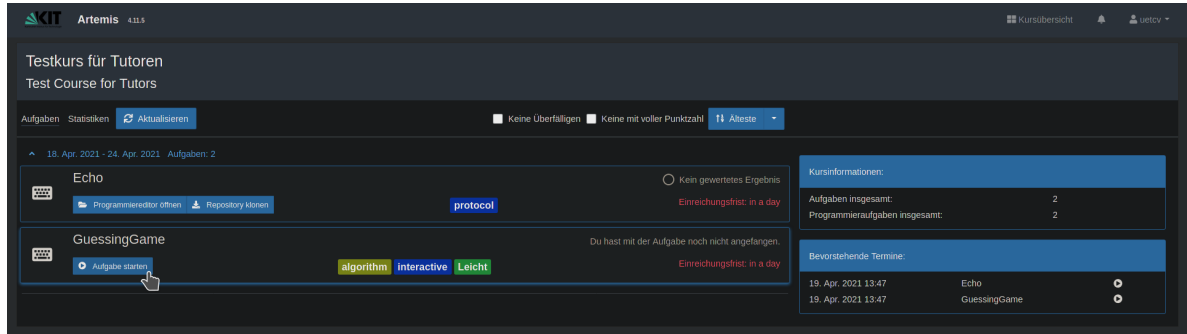
Einrichtung Entwicklungsumgebung  
ooooo

Erstes Java Programm  
ooooo

Typen und Variablen  
ooooo

Klassen und Objekte  
o  
o  
ooooo

# Artemis - Aufgabe starten



Artemis 4.11.5

Kursübersicht

uetcv

Testkurs für Tutoren  
Test Course for Tutors

Aufgaben Statistiken Aktualisieren

Keine Überfalligen Keine mit voller Punktzahl Alteste

18. Apr. 2021 - 24. Apr. 2021 Aufgaben: 2

**Echo** ☐ Kein gewertetes Ergebnis  
 Programmiereditor öffnen Repository klonen protocol Einreichungsfrist: in a day

**GuessingGame** Du hast mit der Aufgabe noch nicht angefangen.  
 Aufgabe starten algorithm interactive Leicht Einreichungsfrist: in a day

**Kursinformationen:**

Aufgaben insgesamt:	2
Programmieraufgaben insgesamt:	2

**Bevorstehende Termine:**

19. Apr. 2021 13:47	Echo	<input type="radio"/>
19. Apr. 2021 13:47	GuessingGame	<input type="radio"/>

## 1. „Aufgabe starten“ anklicken

Organisatorisches  
○○○○○○○○○○

Artemis  
○○●○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

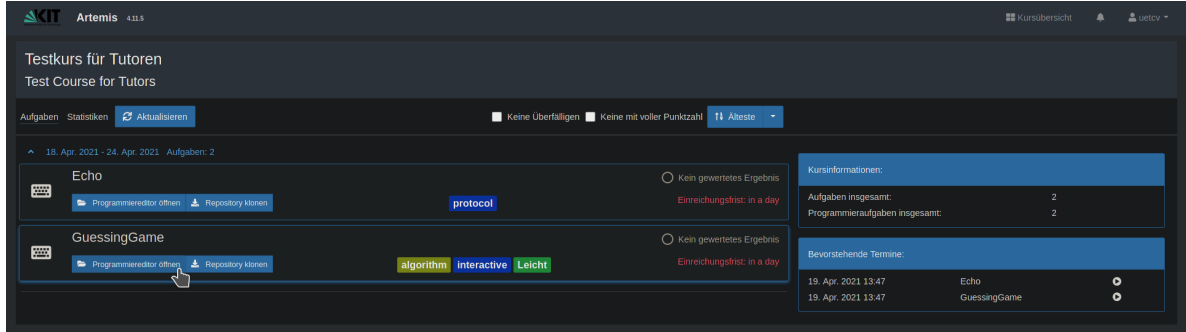
Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○



# Artemis - Webeditor nutzen



Testkurs für Tutoren  
Test Course for Tutors

Aufgaben Statistiken Aktualisieren

Keine Überfalligen Keine mit voller Punktzahl Alteste

18. Apr. 2021 - 24. Apr. 2021 Aufgaben: 2

**Echo** ☐ Kein gewertetes Ergebnis  
Einreichungsfrist: in a day

Programmiereditor öffnen Repository klonen protocol

**GuessingGame** ☐ Kein gewertetes Ergebnis  
Einreichungsfrist: in a day

Programmiereditor öffnen Repository klonen algorithm interactive Leicht

**Kursinformationen:**

Aufgaben insgesamt:	2
Programmieraufgaben insgesamt:	2

**Bevorstehende Termine:**

19. Apr. 2021 13:47	Echo	○
19. Apr. 2021 13:47	GuessingGame	○

## 2. „Programmiereditor öffnen“ anklicken

Organisatorisches  
○○○○○○○○○○



Artemis  
○○○●○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○


**Artemis** 4.11.5
 
 Kursübersicht
 
 uetcv

Echo [5 Punkte]
 


☐ Kein gewertetes Ergebnis
 


Aktualisieren


Speichern

Abenden


Dateibrowser


+ 


+ 



src

 Gespeichert.

 Abgesendet.



Wähle eine Datei aus, um zu beginnen!

Aufgabenstellung

### Echo

Write a program, that reads lines from System.in and outputs the lines again. Remember to send a newline character "ln" after each line or use System.out.println.

Use only java.io, java.lang

The program quits when reading the special input "quit".

- Try to run in all sorts of Timeouts
- Try to use restricted packages

Please report back (eg. Tutor-Forum in Ilias):

- How many tests were you able to see?
- How useful are the error-messages? Please be specific in what you did and what you expect
- Did the tests run as expected?
- Did the system crash? How?

Organisatorisches  
○○○○○○○○○○

Artemis  
○○○○●○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

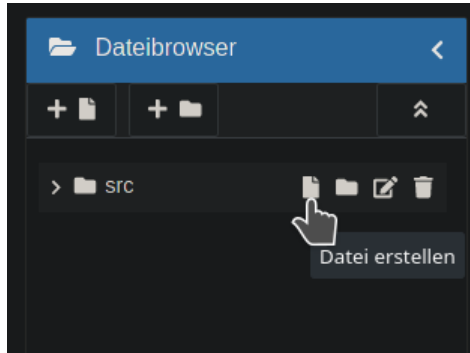
Klassen und Objekte  
○  
○  
○○○○○

18/50

02.11.2022

Péter Bohner: Tutorium 14

Programmieren Tutorium



3. Über den Dateibrowser können neue Dateien angelegt werden
4. Code schreiben

Organisatorisches  
oooooooooooo

Artemis  
oooo●oooooooo

Einrichtung Entwicklungsumgebung  
oooo

Erstes Java Programm  
oooo

Typen und Variablen  
oooo

Klassen und Objekte  
o  
o  
oooo

The screenshot shows the Artemis IDE interface. At the top, there's a header with the KIT logo, the name 'Artemis', and the version '4.11.5'. On the right, there are links for 'Kursübersicht', a notification bell, and a user profile 'uetcv'. Below the header, the main workspace is divided into three panels. The left panel is a 'Dateibrowser' (File Browser) showing a directory structure with 'src' and 'Main.java'. The middle panel is a code editor showing the content of 'src/Main.java', which contains a simple Java program that prints 'hello world!'. The right panel is titled 'Aufgabenstellung' (Task Setup) and contains the task details for 'Echo'. The task description asks the user to write a program that reads lines from 'System.in' and outputs them again. It also lists requirements for using 'java.io', 'java.lang', and restricted packages. At the bottom of the right panel, there are checkboxes for 'Ungespeichert' (Not Saved) and 'Nicht abgesendet' (Not Submitted). The 'Speichern' (Save) button is highlighted by a mouse cursor.

5. „Speichern“ um Änderung zu sichern

6. „Absenden“ um Projekt zu builden, testen und abzugeben

Organisatorisches  
○○○○○○○○○○

Artemis  
○○○○○●○○○○○

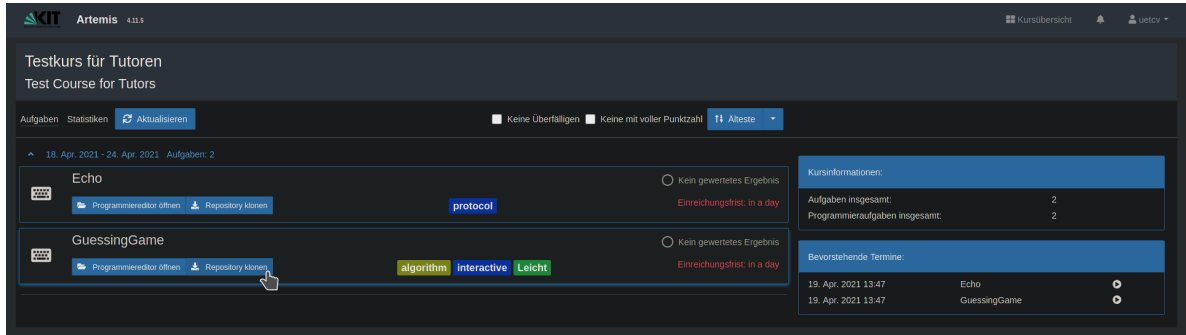
Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

# Artemis - Lokal arbeiten



Artemis 4.11.5

Kursübersicht

uetcv

Testkurs für Tutoren  
Test Course for Tutors

Aufgaben Statistiken Aktualisieren

Keine Überfalligen Keine mit voller Punktzahl Alteste

18. Apr. 2021 - 24. Apr. 2021 Aufgaben: 2

**Echo**

Kein gewertetes Ergebnis

Einreichungsfrist: in a day

Programmieditor öffnen Repository klonen protocol

**GuessingGame**

Kein gewertetes Ergebnis

Einreichungsfrist: in a day

Programmieditor öffnen Repository klonen algorithm interactive Leicht

**Kursinformationen:**

Aufgaben insgesamt:	2
Programmieraufgaben insgesamt:	2

**Bevorstehende Termine:**

19. Apr. 2021 13:47	Echo	○
19. Apr. 2021 13:47	GuessingGame	○

## 2. „Repository klonen“ anklicken

Organisatorisches  
○○○○○○○○○○

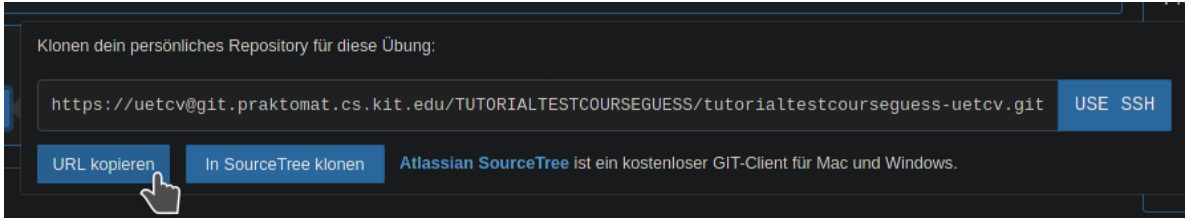
Artemis  
○○○○○○●○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○



3. „URL kopieren“ anklicken
4. eclipse öffnen
5. In eclipse: „File“ → „Import“
6. „Git“ → „Projects from Git“
7. „Next“ klicken

Organisatorisches  
○○○○○○○○○○

Artemis  
○○○○○○○●○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

Import Projects from Git

Source Git Repository

Enter the location of the source repository.

Location

URI:  Local Folder... Local Bundle File...

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:

Password:

☐ Store in Secure Store

? < Back Next > Cancel Finish

8. Unter „URI“ kopierte URL einfügen
9. Unter „Authentication“ Artemis Passwort eintragen
10. 3 x „Next“ klicken
11. „Finish“ klicken

Organisatorisches  
○○○○○○○○○○

Artemis  
○○○○○○○○●○○○

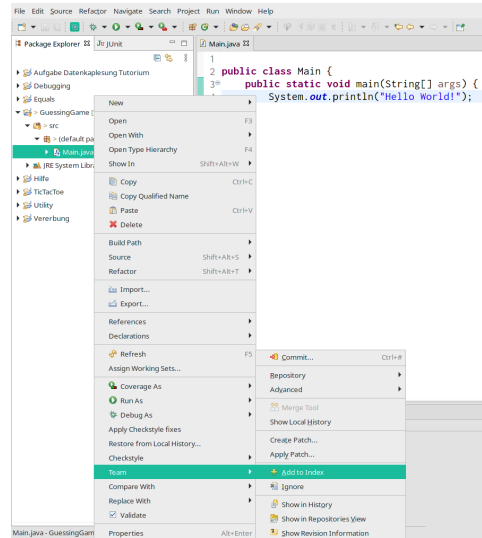
Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

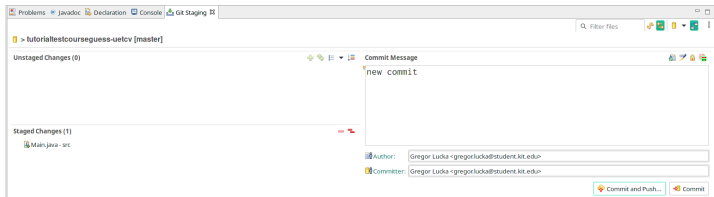
Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

12. Repository ist nun lokal verfügbar
13. Nun kann normal lokal in eclipse programmiert werden
14. geänderte/erstellte Dateien müssen nun „gestaged“ und „committed“ werden
15. „Rechtsklick“ auf geänderte Datei → „Team“ → „Add to Index“
16. wurden alle geänderten Dateien „gestaged“:  
„Rechtsklick“ auf Projekt → „Team“ → „Commit“







17. beliebige „Commit Message“ wählen
18. „Commit and Push...“ klicken
19. „Preview“ klicken
20. „Push“ klicken
21. „Close“ klicken
22. Änderungen sind nun in Artemis eingebucht und die Tests werden ausgeführt
23. Ergebnisse der Tests sind in Artemis verfügbar

Organisatorisches  
○○○○○○○○○○

Artemis  
○○○○○○○○○○●○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

## Fragen?

Organisatorisches  
oooooooooooo

Artemis  
oooooooooooo●

Einrichtung Entwicklungsumgebung  
ooooo

Erstes Java Programm  
ooooo

Typen und Variablen  
oooooo

Klassen und Objekte  
o  
o  
oooooo

# Installation Java Development Kit (JDK)

## Windows/MacOS

Von <https://adoptium.net/> herunterladen und installieren

## Linux

```
sudo apt install openjdk-17-jdk
sudo dnf install java-17-openjdk
sudo pacman -S jdk17-openjdk
# oder sonstiger Paketmanager
```

Nur unter Windows: JDK in PATH eintragen:

Systemumgebungsvariablen bearbeiten → Erweitert → Umgebungsvariablen... → Pfad zum „bin“ Ordner des JDK zu Path hinzufügen

Organisatorisches  
○○○○○○○○○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
●○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

# Installation Git

## Windows/MacOS

Windows/MacOS: Aktuelle Version von <https://git-scm.com/downloads> herunterladen und installieren.

## Debian/Ubuntu

```
sudo apt-get install git
```

Viele Linux Distributionen liefern git bereits aus.

Organisatorisches  
○○○○○○○○○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
●●○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

# Installation Eclipse

## Windows/MacOS

Windows/MacOS: Aktuelle Version von <https://www.eclipse.org/> herunterladen und installieren.

## Debian/Ubuntu

```
sudo apt-get install eclipse
```

Natürlich ist eine andere IDE (z.B. IntelliJ) auch in Ordnung.

Organisatorisches  
○○○○○○○○○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○●○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

# Einrichtung Eclipse

## Leerzeichen statt Tabs

Window → Preferences... → General → Editors → Text Editors → Haken bei „Insert Spaces for Tabs setzen“  
Window → Preferences... → Java → Code Style → Formatter → Edit → Tab policy → Spaces only

## Checkstyle installieren

1. Help → Eclipse Marketplace
2. Nach „Checkstyle“ suchen
3. Bei „Checkstyle Plug-in X.XX.X“ von Lars Ködderitzsch auf „Install“ klicken
4. Lizenz akzeptieren
5. Checkstyle Einstellungen sind nun Global und pro Projekt verfügbar

Organisatorisches  
○○○○○○○○○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○●○○

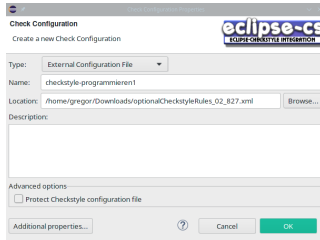
Erstes Java Programm  
○○○○○

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

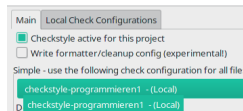
# Einrichtung Checkstyle

Rechtsklick auf Projekt → Checkstyle → Auf Tab „Local Check Configurations“ → Add



1. Bei „Type“ „External Configuration File“ auswählen
2. Name beliebig wählen
3. Bei „Location“ die entsprechende Checkstyle Konfiguration (.xml) auswählen

Auf Tab „Main“ → Haken bei „Checkstyle active for this project“ und Konfiguration auswählen



# Erstes Java Programm

Die *main*-Methode ist der Haupteinstiegspunkt in die Anwendung. Hier startet die Programmausführung.

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

Java Dateien werden **exakt** nach der Klasse benannt: Sonst Compilerfehler.

Organisatorisches  
oooooooooooo

Artemis  
oooooooooooooooo

Einrichtung Entwicklungsumgebung  
ooooo

Erstes Java Programm  
●oooo

Typen und Variablen  
oooooo

Klassen und Objekte  
o  
o  
oooooo



# Kompilieren und Ausführens

Kompilieren einer Datei **JavaApp.java**

```
javac JavaApp.java
```

Ausführen des kompilierten Programms

```
java JavaApp
```

Oder zusammen

```
javac JavaApp.java && java JavaApp
```

Seit Java 11:

```
java JavaApp.java
```

# Kommandozeilenargumente

Argumente an die Java Anwendung mitübergeben:

```
java JavaApp Hallo Welt
```

*args* ist ein Feld (Array) das die Argumente enthält. Also Zugriff mit:

```
args[0]  „Hallo“
args[1]  „Welt“
args[2]  ArrayIndexOutOfBoundsException
```

# Euer erstes Java Programm

## Übung

Schreibt ein Programm das per Kommandozeile einen Vornamen erhält und diesen grüßt.

Beispiel: „Hallo, Peter!“

Nutzt für die Ausgabe die Java Methode `System.out.println(ausgabeString);`

## Zusatz

Was passiert, wenn ihr kein oder mehr als ein Argument übergibt? Probiert es aus...

Organisatorisches  
oooooooooooo

Artemis  
oooooooooooooooo

Einrichtung Entwicklungsumgebung  
ooooo

Erstes Java Programm  
ooo●oo

Typen und Variablen  
oooooo

Klassen und Objekte  
o  
o  
oooooo

# Lösung

```
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hallo, " + args[0] + "!");
    }
}
```

## Zusatz

**Kein Argument:** Wir greifen auf den ersten Eintrag im Feld *args* zu, dieses ist aber leer. Es folgt:

**ArrayIndexOutOfBoundsException: Index 0 out of bounds for length 0** Mehr als ein Argument:

Eingabe:

```
java HelloWorld Peter Sven
```

Ausgabe

```
Hallo, Peter!
```

Das Feld *args* enthält zwar „Sven“ am Index 1, aber unser Programm ruft diesen Eintrag nie auf.

Organisatorisches  
○○○○○○○○○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○●

Typen und Variablen  
○○○○○

Klassen und Objekte  
○  
○  
○○○○○

# Datentypen

Was ist der Unterschied zwischen  $\mathbb{Z}$  und  $\mathbb{Q}$ ?

## Typen

- definiert Menge „gleichartiger“ Werte
- legt mögliche Werte fest, die Variablen, Attribute, Methoden,... annehmen können
- Datentyp legt fest, welche Operationen auf den Werten möglich sind
- in Java acht elementare/primitive Datentypen
  - ganze Zahlen, Fließkommazahlen, Wahrheitswerte, Zeichen
- jede Klasse stellt ebenfalls einen eigenen Datentyp dar (z.B. String)

Organisatorisches  
○○○○○○○○○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
●○○○○○

Klassen und Objekte  
○  
○  
○○○○○

# primitive Datentypen

Typ	Erklärung	Wertebereich	Beispielwerte
boolean	Wahrheitswerte	true oder false	true, false
char	16-Bit-Unicode	0x0000 ... 0xffff	'A', '\n', '\u00D0'
byte	8-Bit-Integer	$-2^7 \dots 2^7 - 1$	12
short	16-Bit-Integer	$-2^{15} \dots 2^{15} - 1$	12
int	32-Bit-Integer	$-2^{31} \dots 2^{31} - 1$	12
long	64-Bit-Integer	$-2^{63} \dots 2^{63} - 1$	12L, 14L
float	32-Bit-Gleitk.	1,40239846E-45f... 3,40282347E+38f	9.81F, 0.3E-8F, 2f
double	64-Bit-Gleitk.	4,94065645841246544E-324... 1,79769131486231570E+308	9.81, 3e1

Organisatorisches  
○○○○○○○○○○

Artemis  
○○○○○○○○○○○○

Einrichtung Entwicklungsumgebung  
○○○○○

Erstes Java Programm  
○○○○○

Typen und Variablen  
○●○○○○

Klassen und Objekte  
○  
○  
○○○○○

Ideen?

## Zeichenkette

- Zusammensetzung aus char mittels String
- Ist kein primitiver Datentyp, sondern eine Klasse
- Außerdem sind ein paar Besonderheiten z.B. beim Vergleichen zu beachten

Organisatorisches  
oooooooo

Artemis  
oooooooooooo

Einrichtung Entwicklungsumgebung  
oooo

Erstes Java Programm  
oooo

Typen und Variablen  
oo●ooo

Klassen und Objekte  
o  
o  
ooooo

# Aufgabe

## Welche Datentypen eignen sich für die folgenden Daten

- Eine Telefonnummer
- Die Erdbeschleunigung in  $m/s^2$
- Vorname und Nachname einer Person
- Alter einer Person in Jahren

## Lösung

- Eine Telefonnummer → String
- Die Erdbeschleunigung in  $m/s^2$  → double (oder float)
- Vorname einer Person → String
- Alter einer Person in Jahren → int

Organisatorisches  
oooooooooooo

Artemis  
oooooooooooooooo

Einrichtung Entwicklungsumgebung  
ooooo

Erstes Java Programm  
ooooo

Typen und Variablen  
ooo●oo

Klassen und Objekte  
o  
o  
ooooo



Eine Variable ist ein „Platzhalter“ für Werte eines Datentyps

## Deklaration

- legt Name und Typ der Variable vor Verwendung fest
- Schema: Typ Name;
- Beispiele
  - **double** a;
  - **int** x, y, z;

Organisatorisches  
oooooooooooo

Artemis  
oooooooooooooooo

Einrichtung Entwicklungsumgebung  
ooooo

Erstes Java Programm  
ooooo

Typen und Variablen  
oooo●o

Klassen und Objekte  
o  
o  
ooooo

# Variablen

## Zuweisung

- setzen eines Wertes
- Schema: `Name = Wert;`
- Beispiele:
  - `a = 2.2;`
  - `x = y;`
  - `x = y + 1;`

## Initialisierung

- Kombination aus Deklaration und Zuweisung
- Schema: `Typ Name = Wert;`
  - Beispiel: `String modul = "Programmieren";`

Organisatorisches  
oooooooo

Artemis  
oooooooooooo

Einrichtung Entwicklungsumgebung  
ooooo

Erstes Java Programm  
ooooo

Typen und Variablen  
oooo●

Klassen und Objekte  
o  
o  
ooooo

# Klassen und Objekte

## Allgemein

- Modellierung der Realität
- Objekte mit Eigenschaften und Verhalten

## Was ist eine Klasse?

- Objekte, die ähnliche Eigenschaften und dasselbe Verhalten haben, werden zusammengefasst
- Klasse stellt Bauplan für Objekte dar
- Attribute und Methoden werden festgelegt

Organisatorisches  
oooooooooooo

Artemis  
oooooooooooooooo

Einrichtung Entwicklungsumgebung  
ooooo

Erstes Java Programm  
ooooo

Typen und Variablen  
oooooo

Klassen und Objekte  
●  
○  
oooooo

# Was ist ein Objekt?

## Instanzen, die aus dem Bauplan entstehen können

### Charakterisierung eines Objekts

- **Identität:** Bleibt immer gleich, egal ob sich der Zustand ändert
- **Zustand:** Menge aller Attributwerte
- **Verhalten:** Methoden

Organisatorisches  
oooooooo

Artemis  
oooooooooooo

Einrichtung Entwicklungsumgebung  
oooo

Erstes Java Programm  
oooo

Typen und Variablen  
ooooo

Klassen und Objekte  
o  
●  
ooooo

Wie bauen die Klasse **Laptop**:

## Eigenschaften

- Arbeitsspeicher
- Tastatur
- Display
- Ist es ein Convertible?

## Verhalten

- Hochfahren
- Aufklappen
- Programm ausführen

Organisatorisches  
oooooooo

Artemis  
oooooooooooo

Einrichtung Entwicklungsumgebung  
oooo

Erstes Java Programm  
oooo

Typen und Variablen  
ooooo

Klassen und Objekte  
o  
o  
●oooo

## Attribute

```
class Laptop {  
    int ram;  
    Keyboard keyboard;  
    Display display;  
    boolean isConvertible;  
  
    void powerOn() { ... }  
    void open(){ ... }  
    void runProgram(String name) { ... }  
}
```

# Wie erzeugt man ein Objekt?

Der Ausdruck `new Classname();` erzeugt ein neues Objekt.

Mit Laptop `laptop = new Laptop();` wird das Objekt `laptop` vom Typ `Laptop` erstellt.

`laptop.ram = 16;` ändert den Wert des Arbeitsspeichers.

Mehr zu Konstruktoren kommt bald.

Organisatorisches  
oooooooooooo

Artemis  
oooooooooooooooo

Einrichtung Entwicklungsumgebung  
ooooo

Erstes Java Programm  
ooooo

Typen und Variablen  
oooooo

Klassen und Objekte  
o  
o  
oo●ooo

## Teil A

Modelliert eine Klasse Motor. Ein Motor besitzt eine gewisse Anzahl an PS und Zylindern. Er hat außerdem ein Gewicht und Hubraum (in l). Wählt geeignete Datentypen.

## Teil B

Modelliert eine Klasse Karosserie. Sie hat ein Gewicht, Anzahl an Sitzplätzen und eine bestimmte Form (Name der Form).

## Teil C

Überlegt euch, wie man aus den beiden gegebenen Klassen in Java-Code eine Klasse Auto entwerfen könnte. Zur Info: Ein Auto hat eine Karosserie und einen Motor.

Organisatorisches  
oooooooooooo

Artemis  
oooooooooooooooo

Einrichtung Entwicklungsumgebung  
ooooo

Erstes Java Programm  
ooooo

Typen und Variablen  
ooooo

Klassen und Objekte  
o  
o  
ooo●oo



## Teil A und B

```
class Engine {  
    int horsepower;  
    int numberOfCylinders;  
    int weight;  
    double engineDisplacement;  
}
```

```
class Body {  
    int weight;  
    int numberOfSeats;  
    String type;  
}
```

## Teil C

```
class Car {  
    Body body;  
    Engine engine;  
    ...  
}
```

- Klassen können als eigener Datentyp genutzt werden
- Attribute können also als Typ eine bestimmte Klasse haben
- Sie nutzen deren Eigenschaften und Verhalten

Organisatorisches  
oooooooooooo

Artemis  
oooooooooooooooo

Einrichtung Entwicklungsumgebung  
ooooo

Erstes Java Programm  
ooooo

Typen und Variablen  
oooooo

Klassen und Objekte  
o  
o  
oooo●o

# Bis zum nächsten Tutorium am 09.11.2022!

Organisatorisches  
oooooooooooo

Artemis  
oooooooooooooooo

Einrichtung Entwicklungsumgebung  
ooooo

Erstes Java Programm  
ooooo

Typen und Variablen  
oooooo

Klassen und Objekte  
o  
o  
ooooo●