

## Übungsblatt 1: Pseudocode und Programmcode

Deadline: 16.10.2025 12:00 Uhr

Besprechung: 20.10.–24.10.2025

### Aufgabe 1      Abgabe über GATE; 1c) mit Peer-Review Pseudocode und Programmcode

- a) Nennen Sie eine Gemeinsamkeit und einen Unterschied zwischen Pseudocode und Programmcode.
- b) Gegeben ist folgender Algorithmus in Pseudocode:

**Eingabe:** Natürliche Zahl  $n$

**Ausgabe:** Natürliche Zahl  $x$

Setze  $x := 0$

Setze  $i := 1$

Solange  $i \leq n$  führe aus:

    Falls  $i$  ungerade, dann:

        Addiere  $i$  zu  $x$

    Erhöhe  $i$  um 1

Gib  $x$  aus

Was berechnet der Algorithmus?

- c) Schreiben Sie einen Pseudocode, der für jede Eingabe einer natürlichen Zahl  $n > 0$ , deren Fakultät

$$n! := 1 \cdot 2 \cdot 3 \cdot 4 \cdot \dots \cdot (n-2) \cdot (n-1) \cdot n$$

berechnet.

### Aufgabe 2      freiwillige Abgabe in GATE Compiler und Interpreter

- a) Welche der folgenden Aussagen zu Compilern sind korrekt?
  - i) Compiler können nur syntaktisch korrekten Programmcode kompilieren.
  - ii) Compiler können nur semantisch korrekten Programmcode kompilieren.

- iii) Compiler erzeugen Byte- oder Maschinencode.
  - iv) Aus Programmcode erzeugter Maschinencode ist plattformunabhängig.
  - v) Java ist eine kompilierte Programmiersprache.
  - vi) Kommentare werden vom Compiler ignoriert.
- b) Welche der folgenden Aussagen zu Interpretern sind korrekt?
- i) Byte- oder Maschinencode dient als Eingabe für einen Interpreter.
  - ii) (Menschenlesbarer) Programmcode dient als Eingabe für einen Interpreter.
  - iii) Interpreter sind plattformunabhängig.
  - iv) Java ist eine interpretierte Programmiersprache.

### **Aufgabe 3**      freiwillige Abgabe in GATE                   **Einrichten der IDE und erstes Programm**

Ziel dieser Aufgabe ist es zunächst den das Java JDK 21 auf Ihrem Computer zu installieren, zu testen und schließlich die Entwicklungsumgebung (IDE) Eclipse aufzusetzen. Eine Entwicklungsumgebung unterstützt den Programmierer bei der Entwicklung von Programmen auf unterschiedlichste Arten. Dazu gehört das automatische Kompilieren während der Programmeingabe, das Anzeigen von Fehlern und Verbesserungsvorschlägen während der Programmeingabe, die Generierung von Code, Autovervollständigung und mehr. Befolgen Sie hierzu die folgenden Anweisungen:

- Installation mit Windows
  - i) Laden Sie [Liberica JDK 21 in der Standard Edition](https://bell-sw.com/pages/downloads/) (<https://bell-sw.com/pages/downloads/>) herunter, und installieren Sie es.
  - ii) Downloaden und installieren Sie Eclipse: <https://www.eclipse.org/downloads/>. Wählen Sie bei der Installation "Eclipse IDE for Java Developers" aus.

- Installation mit MacOS

Laden Sie Eclipse herunter  
(<https://www.eclipse.org/downloads/packages/>). Öffnen Sie den Installer und befolgen Sie die Anweisungen.

Alternativ empfehlen wir die Installation mit brew:

```
brew install --cask eclipse-java
```

- Installation mit Linux (Ubuntu)

- i) Installieren Sie Java JDK 21 in der Standard Edition (SE), indem sie den Terminal öffnen (Ctrl-Alt-T) und die folgende Zeile ausführen:

```
sudo apt install openjdk-21-jdk
```

Notfalls geht auch Java JDK 16/17 (z. B. openjdk-17-jdk).

- ii) Installieren Sie weiter [Eclipse](#), indem Sie die folgende Zeile ausführen:

```
sudo snap install --classic eclipse
```

Nun geht es erst einmal darum, Java auf der Kommandozeile zum Kompilieren und Ausführen zu benutzen:

- a) Laden Sie das Programm `HelloWorld.java` von GATE herunter. Öffnen Sie das Terminal und navigieren Sie mit dem Terminal in den entsprechenden Ordner. Kompilieren Sie das Programm mit folgendem Befehl.

```
javac HelloWorld.java
```

Auf Windows können Sie das Terminal direkt im richtigen Ordner öffnen, indem sie im Windows-Explorer die Shift-Taste drücken und den Ordner mit der rechten Maustaste anklicken. Im Kontextmenü erscheint dann der Befehl "Eingabeaufforderung hier öffnen" (Win7) oder "Power Shell-Fenster hier öffnen" (Win10). Alternativ können sie den Ordnerpfad anklicken, "cmd" eingeben und Enter drücken.

Bestätigen Sie die Befehle jeweils mit der Enter-Taste.

- b) Führen Sie das Programm mit dem folgenden Befehl aus:

```
java HelloWorld
```

- c) Nutzen Sie das Tutorial mit einer Anleitung zu Eclipse und einer Anleitung für ein erstes Programm [am Ende des Übungsblattes](#), um ein Programm `ErstesProgramm.java` zu erstellen, welches die Worte "Mein erstes Java Programm" (ohne Anführungszeichen) auf der Konsole ausgibt.

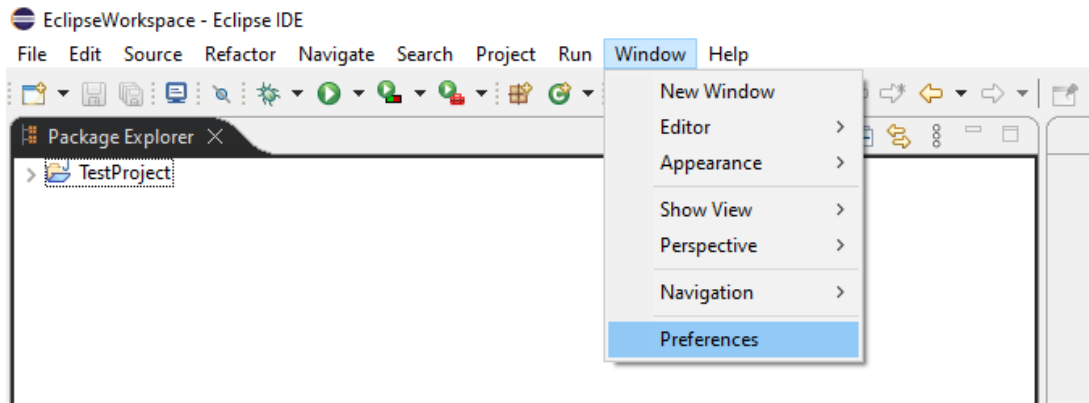
## Abgabe

Die Abgabe erfolgt bis zum 16.10.2025 12:00 Uhr auf GATE, <https://gate.ifi.lmu.de>.

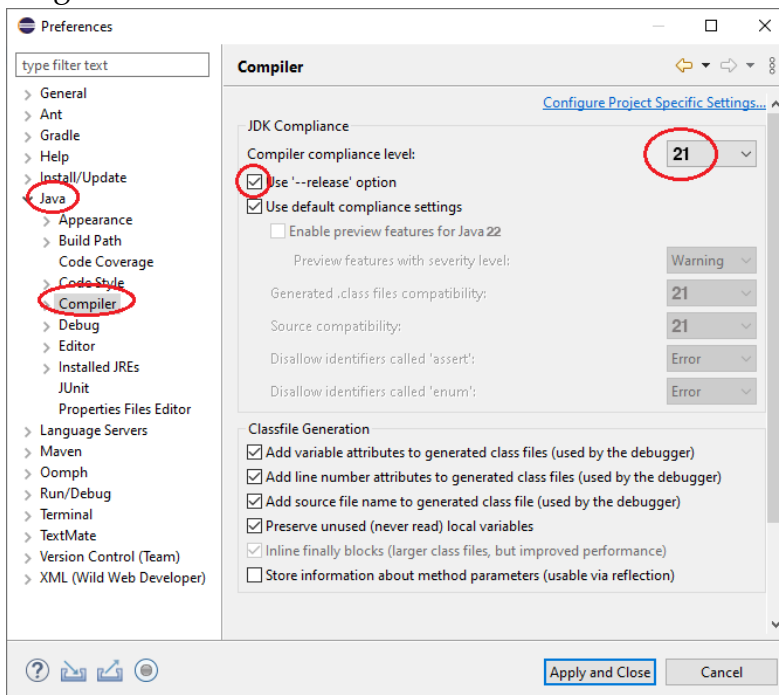
*Zuletzt aktualisiert: 30. September 2025*

## How To Eclipse

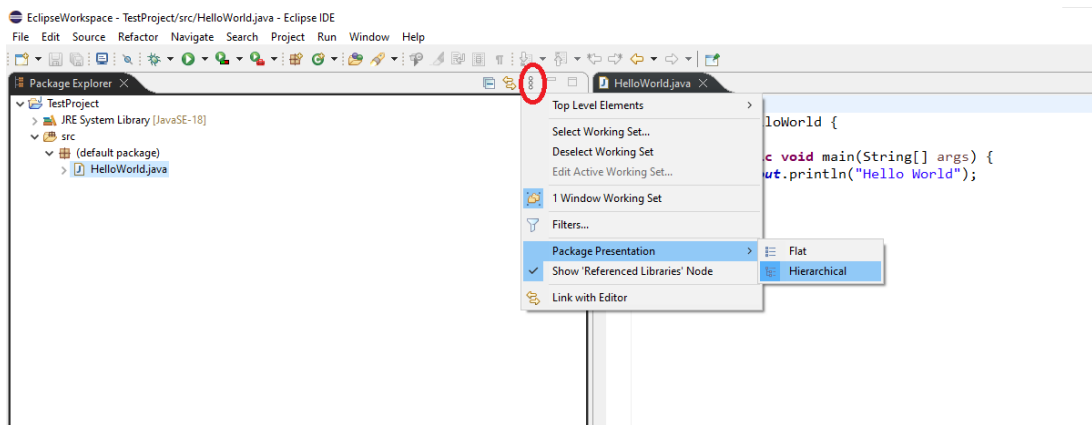
- Die allermeisten Einstellungen für Eclipse können im Menü Window→Preferences vorgenommen werden. (Auf Mac: Eclipse→Settings)



- Damit Eclipse die richtige Java-Version verwendet, sollte folgende Einstellung vorgenommen werden:



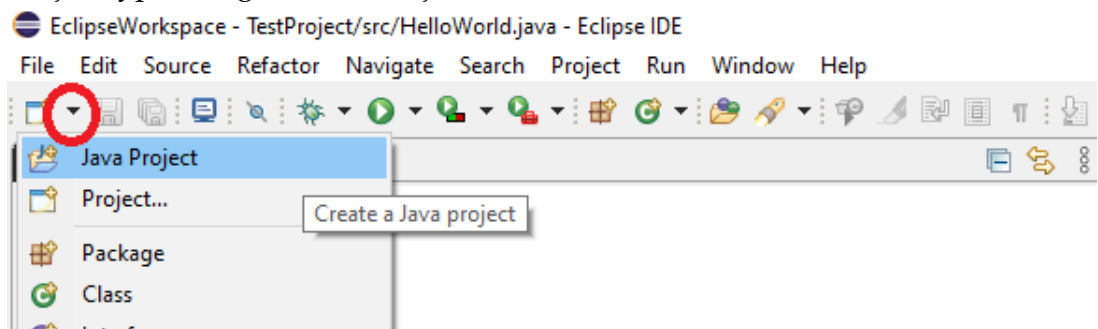
- Anpassen des Package Explorers: Bei hoher Paket-Verschachtelungstiefe liefert *Hierarchical* eine übersichtlichere Darstellung als *Flat*



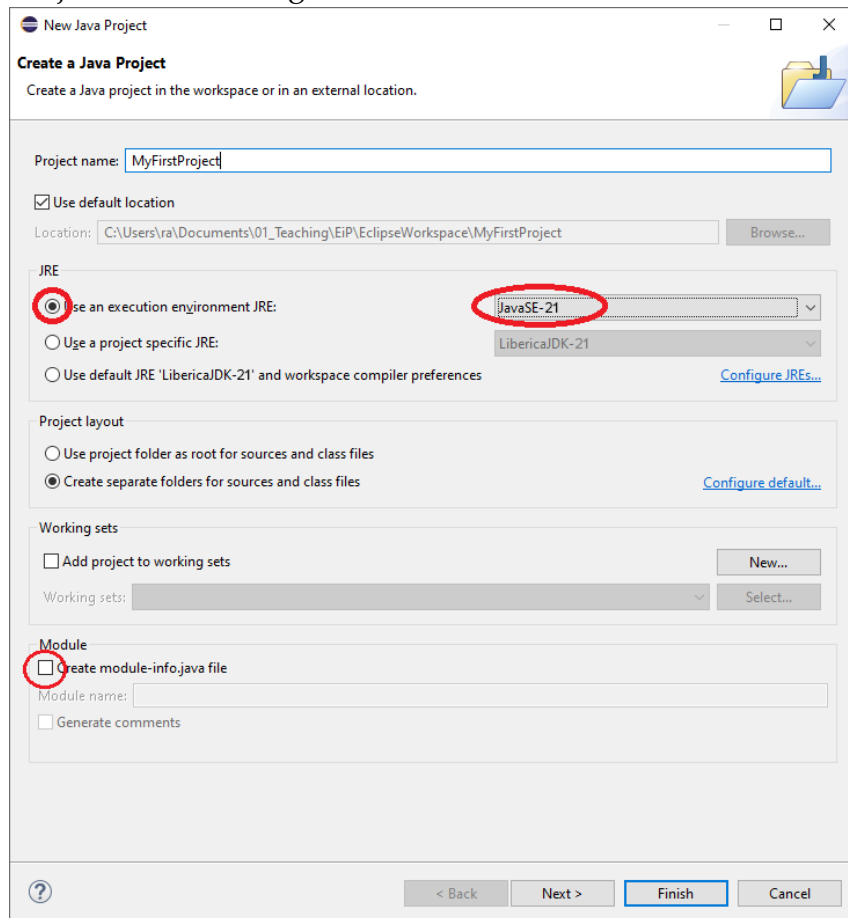
- Zusätzlich zu den globalen Eclipse-Einstellungen kann jedes Projekt noch einmal eigenständig konfiguriert werden.
  - In Eclipse ist der Quellcode in Java-Projekten organisiert.
  - Jede Java-Datei gehört in der Regel einem Java-Projekt an.
- Zum Ändern der Projekteinstellungen muss zunächst das entsprechende Projekt links im Package Explorer markiert sein. Dann können über Project->Properties die Einstellungen geändert werden.

## Ein erstes Java-Programm in Eclipse

- Neues Java-Projekt anlegen
  - Projekttyp festlegen (Java Projekt)

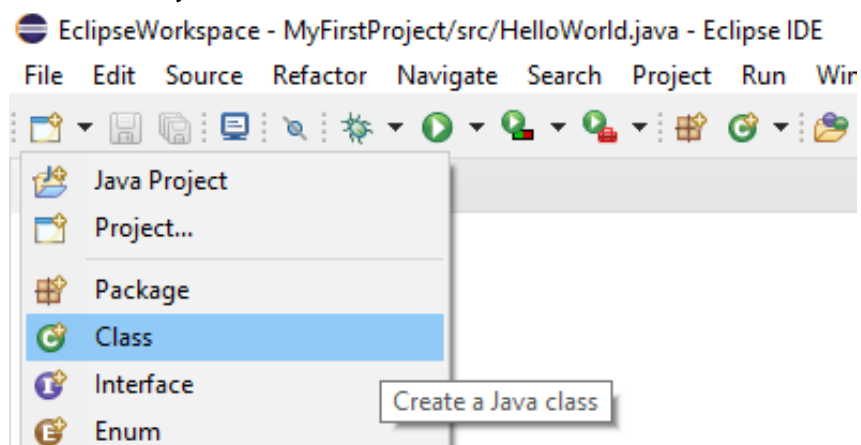


## ii) Projekt-Name festlegen



## b) Klasse hinzufügen

### i) Menü New Java Class



- ii) Klassenname festlegen
- iii) Geben Sie kein Package an, sondern nutzen Sie das default-Package
- iv) Auf Button Finish klicken und die Klasse wird erzeugt

New Java Class

Java Class

The use of the default package is discouraged.

Source folder: MyFirstProject/src Browse...

Package: (default) Browse...

☐ Enclosing type: Browse...

Name: HelloWorld

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static  
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?  
☐ public static void main(String[] args)  
☐ Constructors from superclass  
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

Finish Cancel

c) Die Klasse wird automatisch im Hintergrund kompiliert. Das kann konfiguriert werden mittels Project→Build Automatically

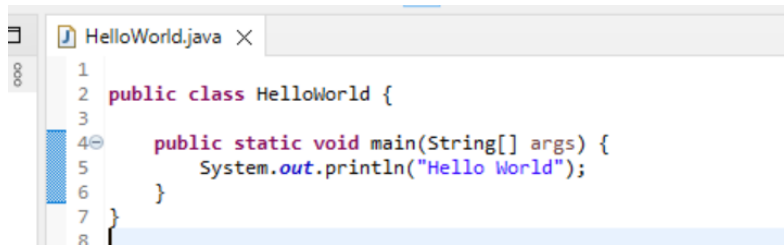
d) Einfügen des Codes und speichern

- Ziel: Erstellen der Main-Methode

```
public static void main(String[] args) {  
  
}
```

- Ziel: Ausgabe von „Hello World“ auf der Konsole

```
System.out.println("Hello, World!");
```



e) Ausführen des Beispiels durch Klick auf den Pfeil neben dem Run-Symbol

