



Programmierung 2

Vorlesung 1: Eclipse-Basics

April 2025

`alexander.gepperth@cs.hs-fulda.de`



Eclipse



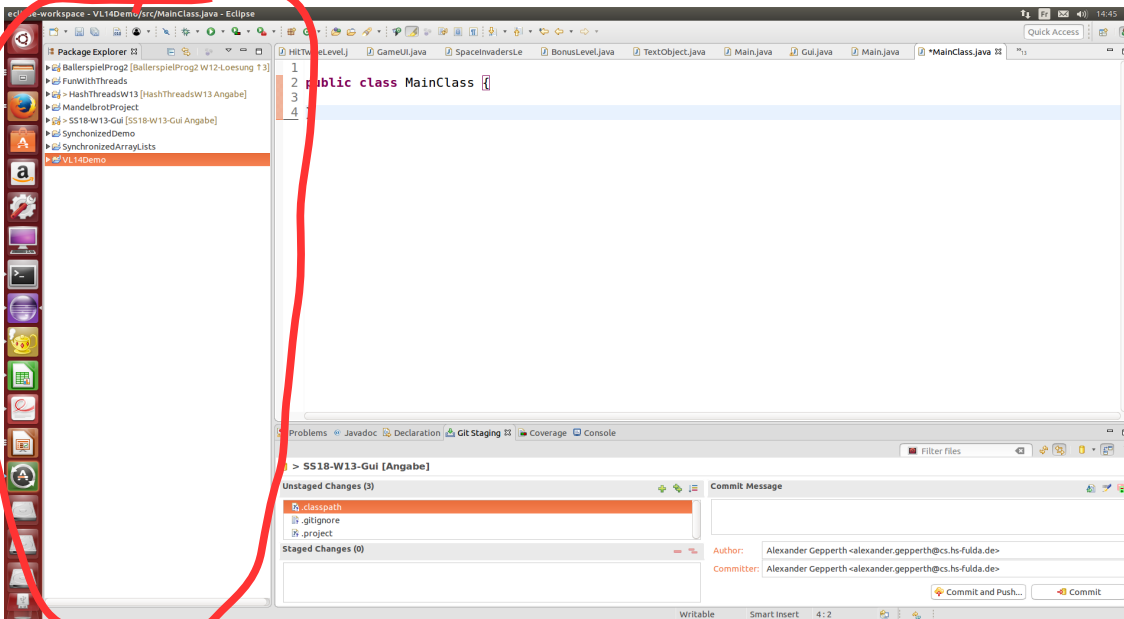
Eclipse

- Stellvertretend für alle IDEs
- Wird in dieser Vorlesung zur Demonstration und für Übungsaufgaben genutzt
- andere sind genauso gut: Visual Studio Code, NetBeans IDE, ...



Der Eclipse-Workspace

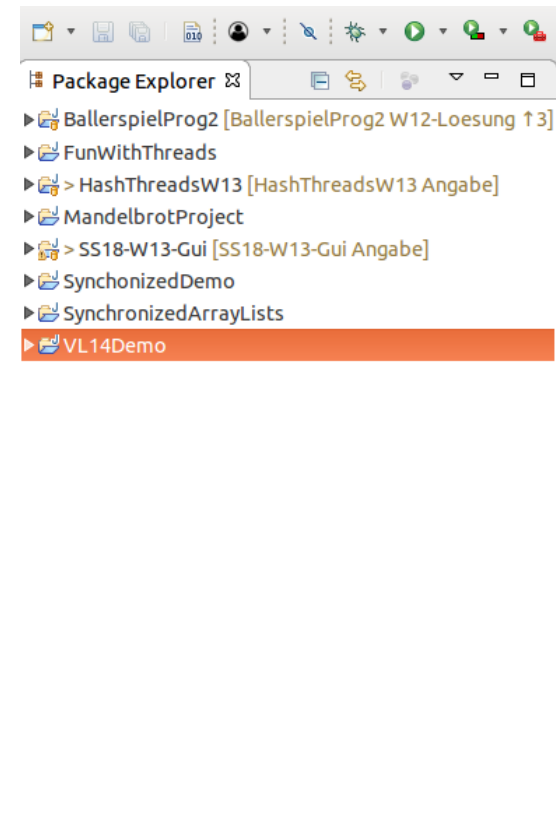
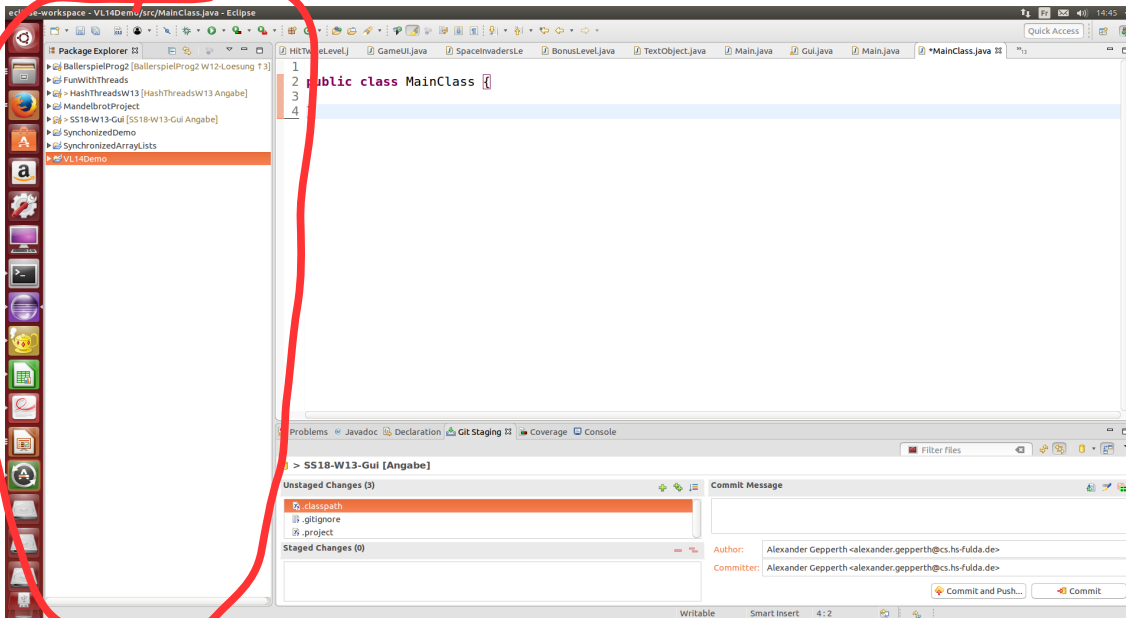
- Workspace = Sammlung aller eigenen Projekte





Der Eclipse-Workspace

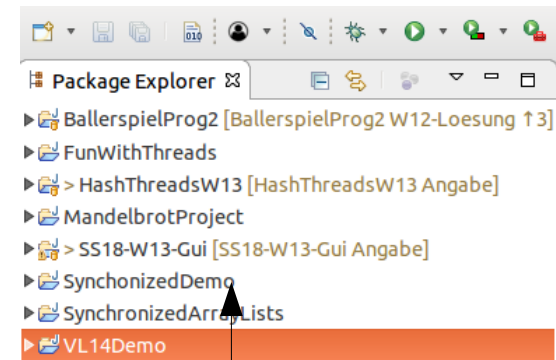
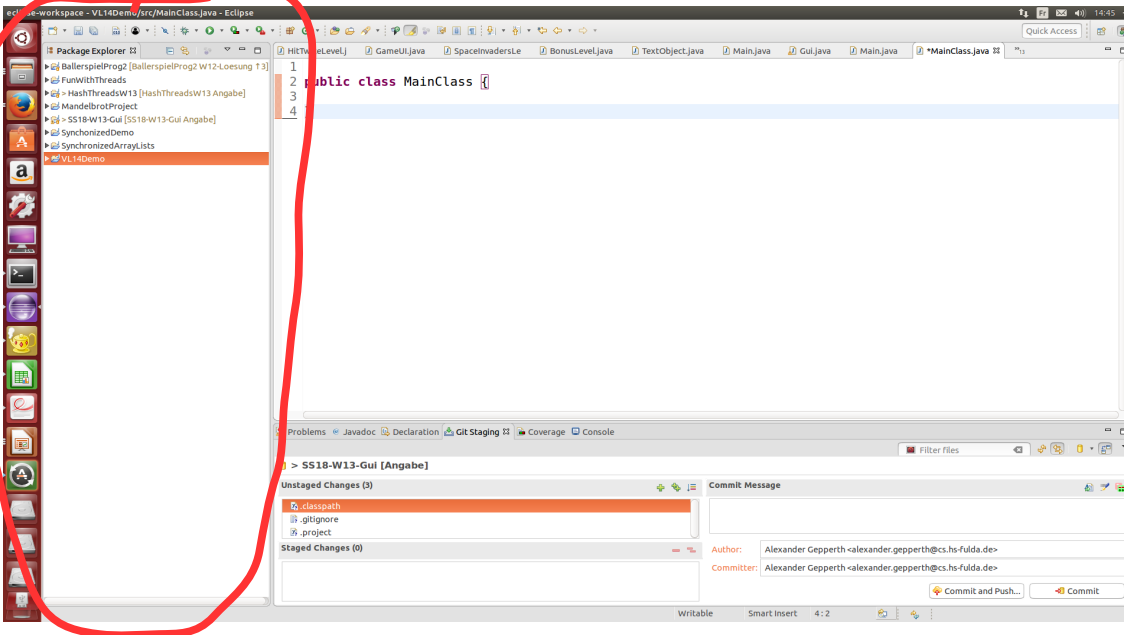
- Workspace = Sammlung aller eigenen **Projekte**





Der Eclipse-Workspace

- Workspace = Sammlung aller eigenen **Projekte**

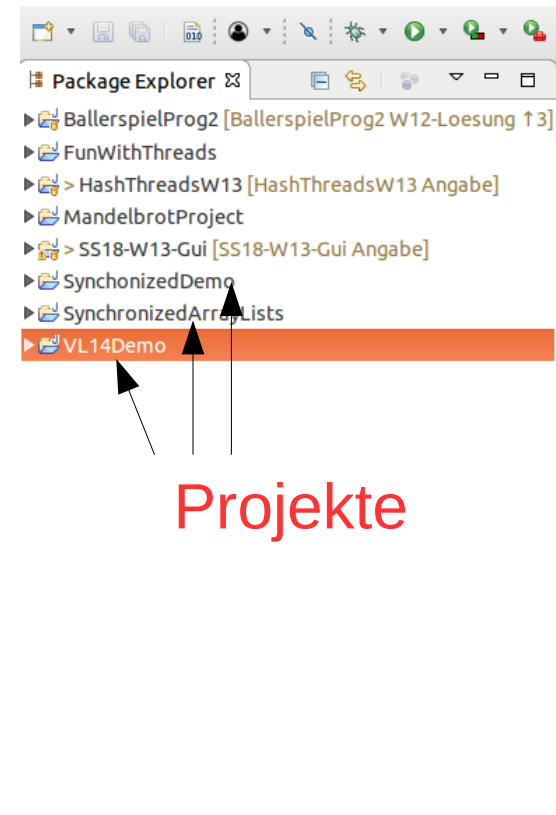
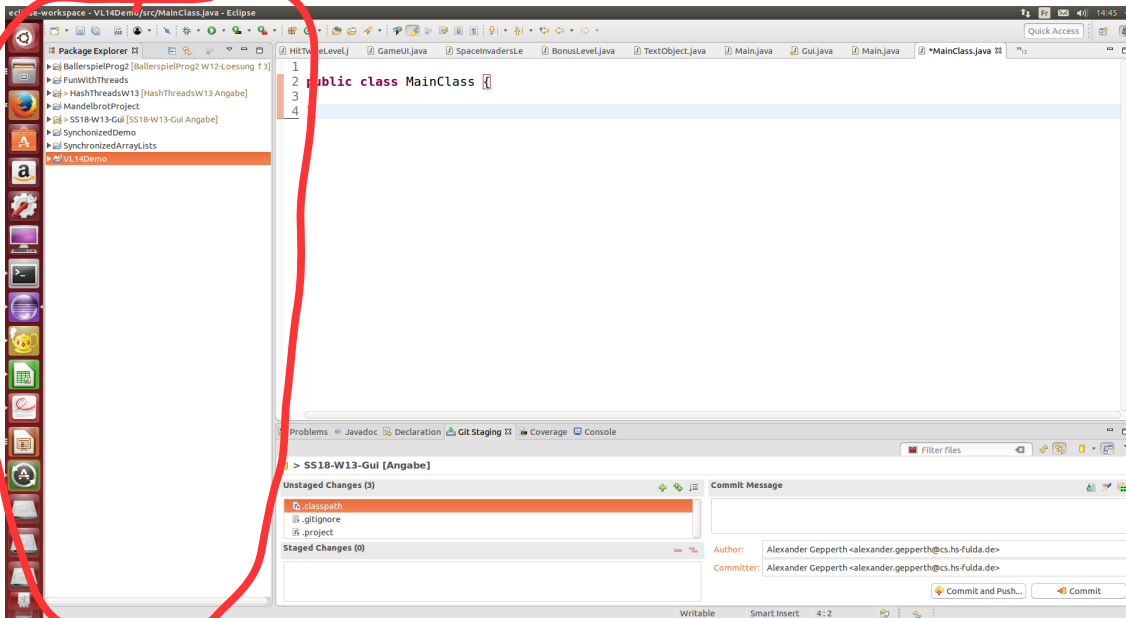


Projekte



Eclipse-Projekte

- **Projekt** ist meist ein ausführbares Programm (muss aber nicht)

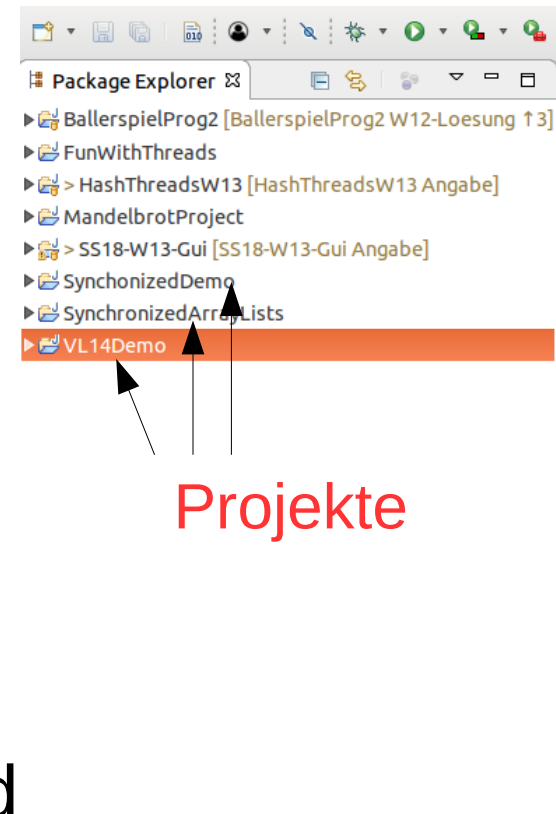


Projekte



Eclipse-Projekte

- **Projekt** ist meist ein ausführbares Programm (muss aber nicht)
- Sammlung von **Klassen** in einer oder mehreren **Packages**
- zusätzliche Datenfiles, alles was nötig ist um das Projekt erfolgreich auszuführen (laufenzulassen)
- Einstellungen zum Compilieren und Ausführen

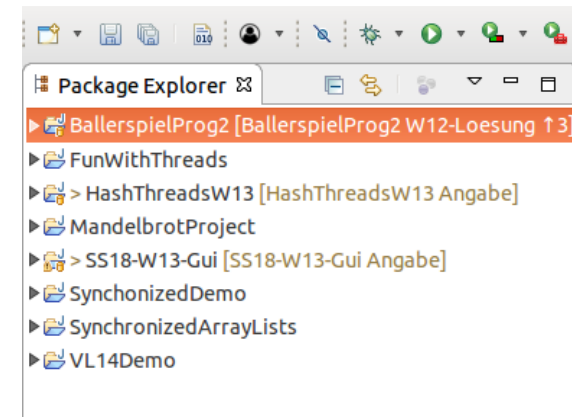


Projekte



Eclipse-Projekte

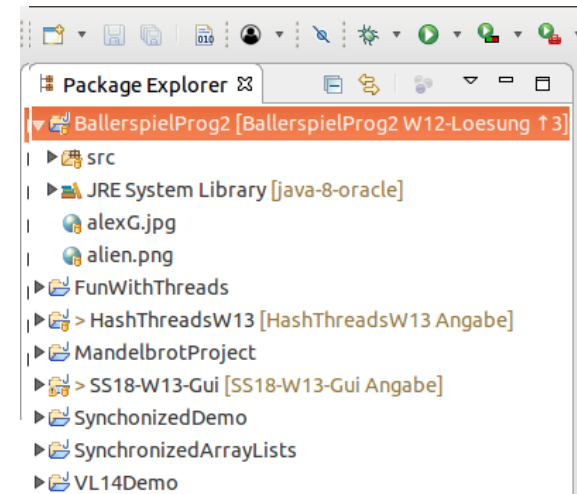
- **Projekt** ist meist ein ausführbares Programm (muss aber nicht)
- Sammlung von **Klassen** in einer oder mehreren **Packages**
- zusätzliche Datenfiles, alles was nötig ist um das Projekt erfolgreich auszuführen (laufenzulassen)
- Einstellungen zum Compilieren und Ausführen





Eclipse-Projekte

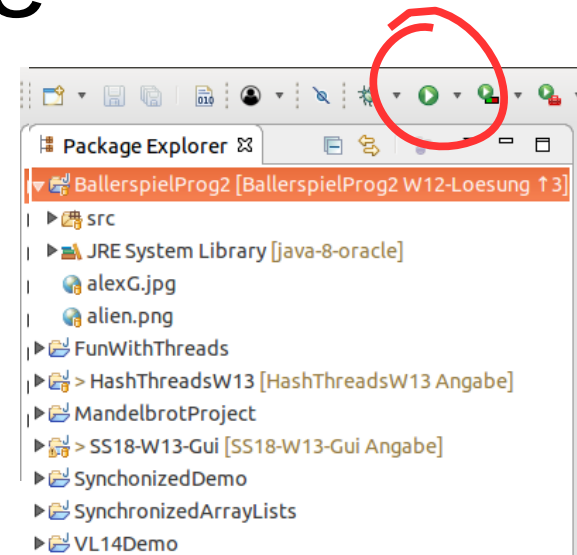
- Eclipse-Java-Projekte sind als Baumstruktur organisiert
- Ebenen/Elemente der Baumstruktur
 - Verzeichnisse
 - Packages
 - Java-Dateien
 - Klassen
 - » Methoden





Eclipse-Projekte

- Eclipse-Projekte die korrekt konfiguriert sind, kann man direkt ausführen!
- Konsolenausgabe kann ebenfalls in Eclipse beobachtet werden!





Eclipse-Projekte: Konsolenausgabe

```
1/** Initial state:[]
38import java.lang.Runnable ;[]
41
42 // hier wird eine Klasse benötigt die Runnable implementiert!
43 class BackgroundCode implements Runnable {
44
45 @Override
46 public void run() {
47     while(true) {
48         if (Main.userInput.equals("Ping")) {
49             System.out.println("Exiting!");
50             System.exit(0);
51         }
52     }
53 }
54 }
55
56 }
57
58 public class Main {
59
60     static volatile String userInput = "";
61
62 }
```

Problems Javadoc Declaration Git Staging Coverage Console
<terminated> Main [Java Application] /usr/lib/jvm/java-8-oracle/bin/java (Jul 3, 2018, 2:57:56 PM)
Enter text:
Test
Enter text:
Ping
Enter text:
Exiting!



Top Eclipse-Features

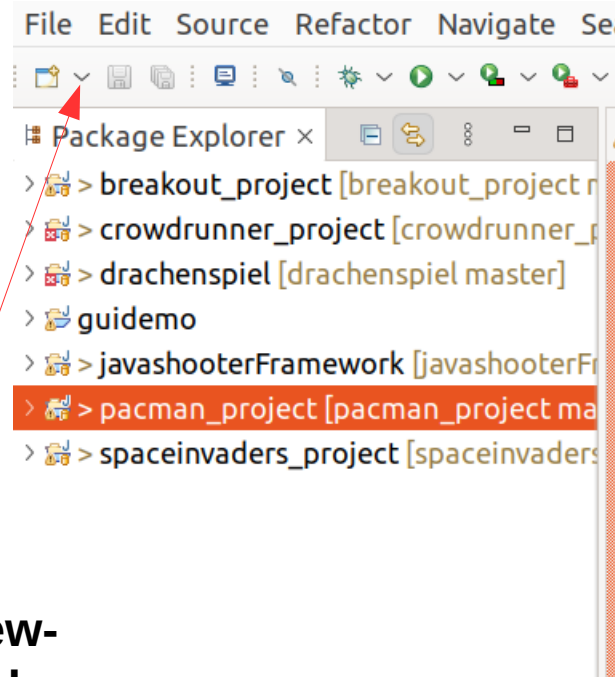
- Fehlererkennung:
rotes Kreuz am
Rand!!
- MouseOver auf
Kreuz gibt
Hilfestellung!
- aber: selbst Nachdenken ist sinnvoll!

```
1
2 class TesteAufTeilbarkeit {
3
4 public static void main(String[] args) {
5 // TODO Auto-generated method stub
6
7 System.out.println("Alex" ;
8
9 }
10
11 }
12 |
```





Neues Java-Projekt erstellen



2. Pfeil neben New-Schaltfläche anklicken,
„Java Project“
auswählen



Neues Java-Projekt erstellen

Create a Java Project
Enter a project name.

Project name:

☒ Use default location
Location: [Browse...](#)

JRE

☒ Use an execution environment JRE:
☐ Use a project specific JRE:
☐ Use default JRE 'jre' and workspace compiler preferences [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files
☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

☐ Add project to working sets [New...](#)
Working sets: [Select...](#)

Module

☒ Create module-info.java file
Module name:
☐ Generate comments

[?](#) [< Back](#) [Next >](#) [Cancel](#) [Finish](#)

Projektname

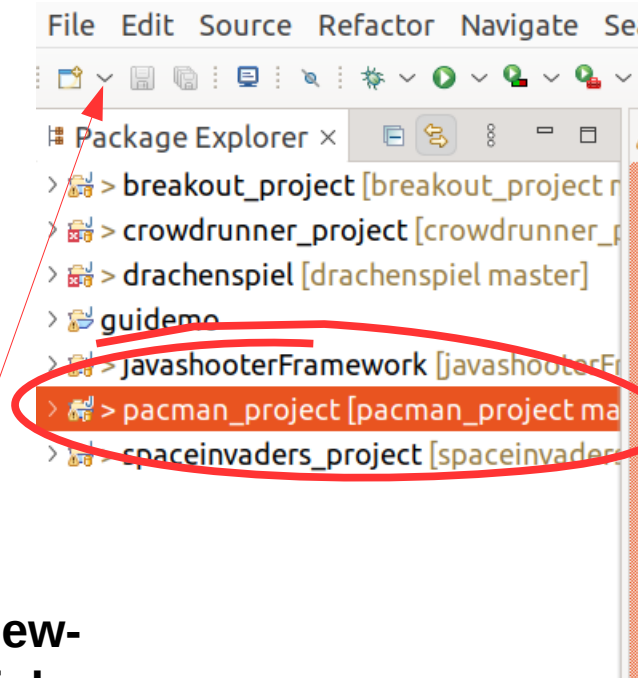
Separater src-Ordner

KEIN module-info.java





Erstellen einer neuen Klasse im Projekt



2. Pfeil neben New-Schaltfläche anklicken, „Class“ auswählen

1. Projekt selektieren



Erstellen einer neuen Klasse im Projekt

The screenshot shows the Eclipse IDE with the 'New Java Class' dialog box open. The 'Package' field is highlighted with a red circle and an arrow pointing to the text 'Name wählen'. The 'Superclass' field is highlighted with a red circle and an arrow pointing to the text 'Oberklasse wählen'. The 'main()' method option is highlighted with a red circle and an arrow pointing to the text 'main() automatisch erzeugen lassen?'. The 'Access' field is also highlighted with a red circle.

Name wählen

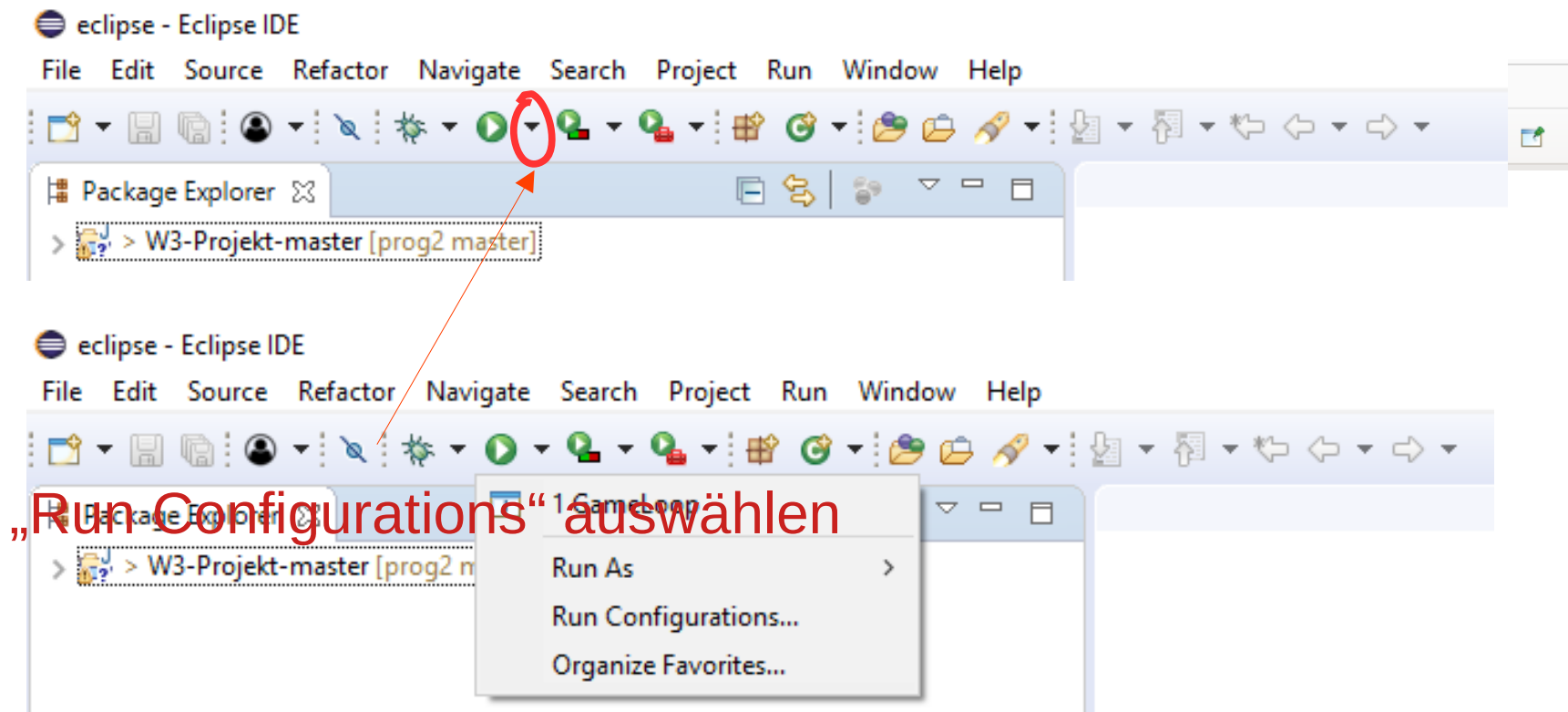
Oberklasse wählen

main() automatisch erzeugen lassen?



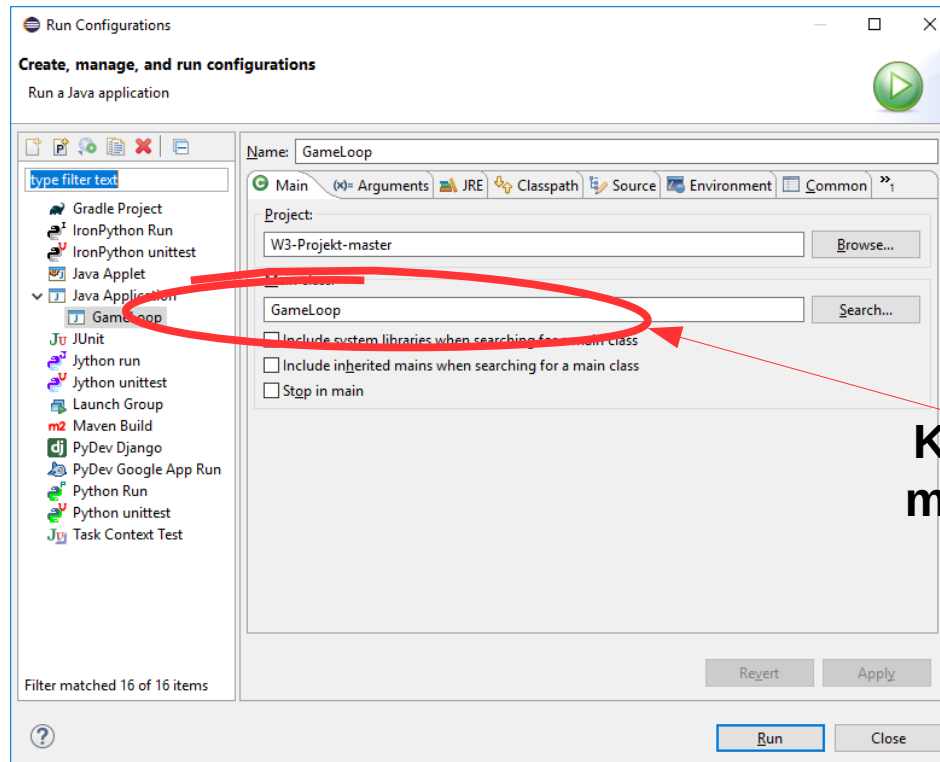


Selektieren welche Klasse ausgeführt wird





Selektieren welche Klasse ausgeführt wird



**Klasse eintragen deren
main(..) ausgeführt werden
soll**



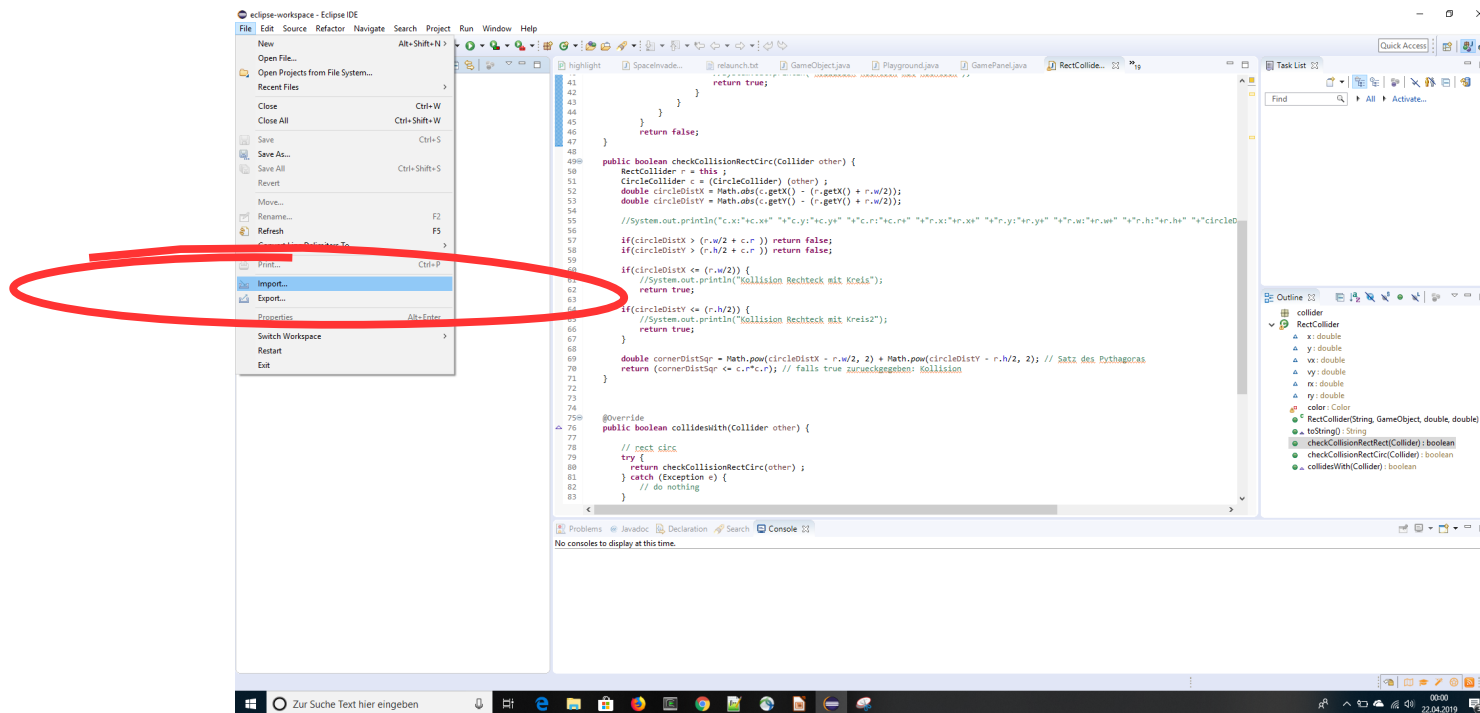
Woche 2: bis hierher!



Advanced features

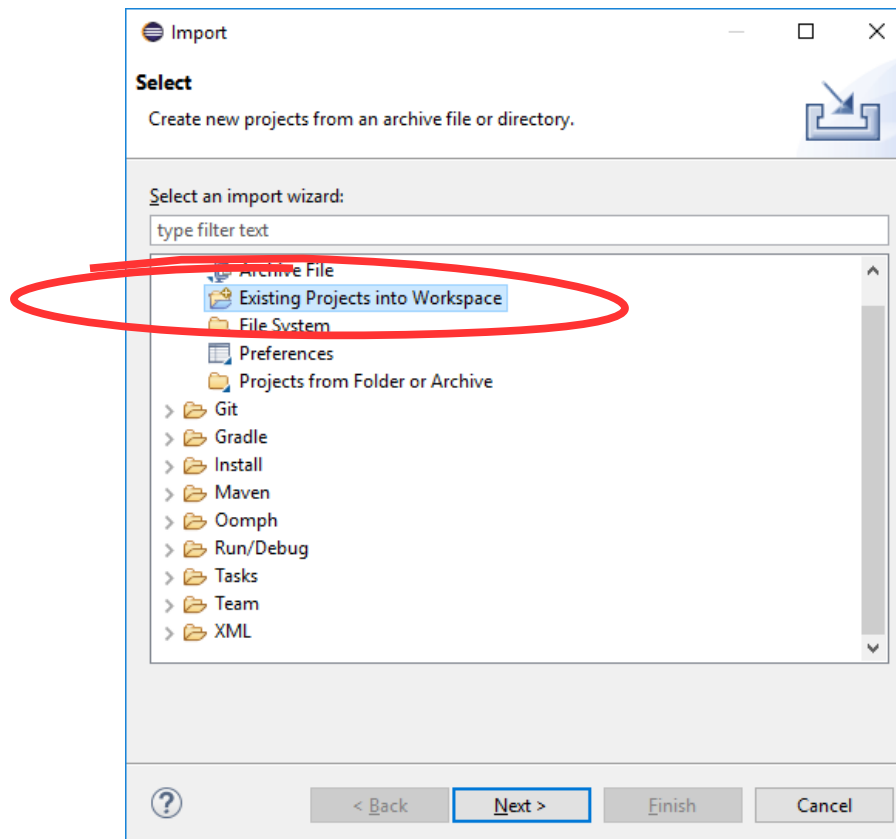


Import von gezippten Projekten



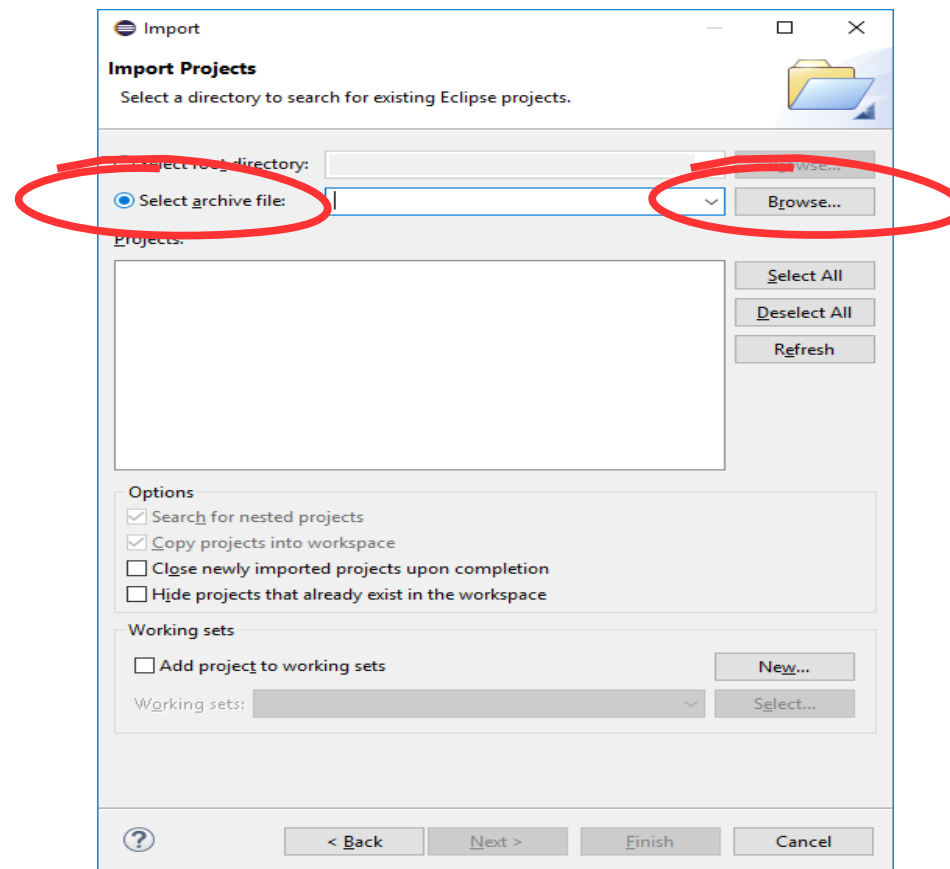


Import von gezippten Projekten





Import von gezippten Projekten



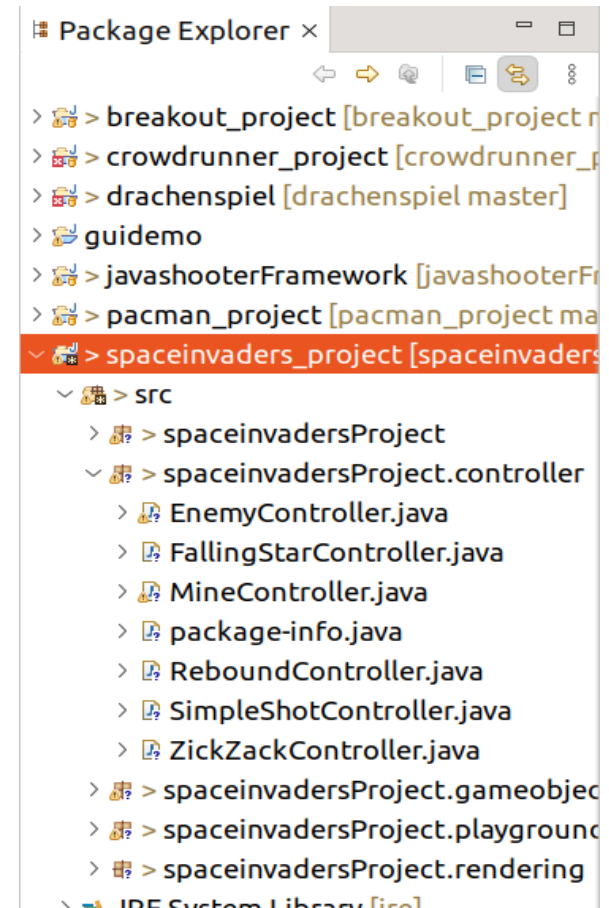
ZIP-Datei selektieren





Abhängigkeiten eines Projekts verwalten

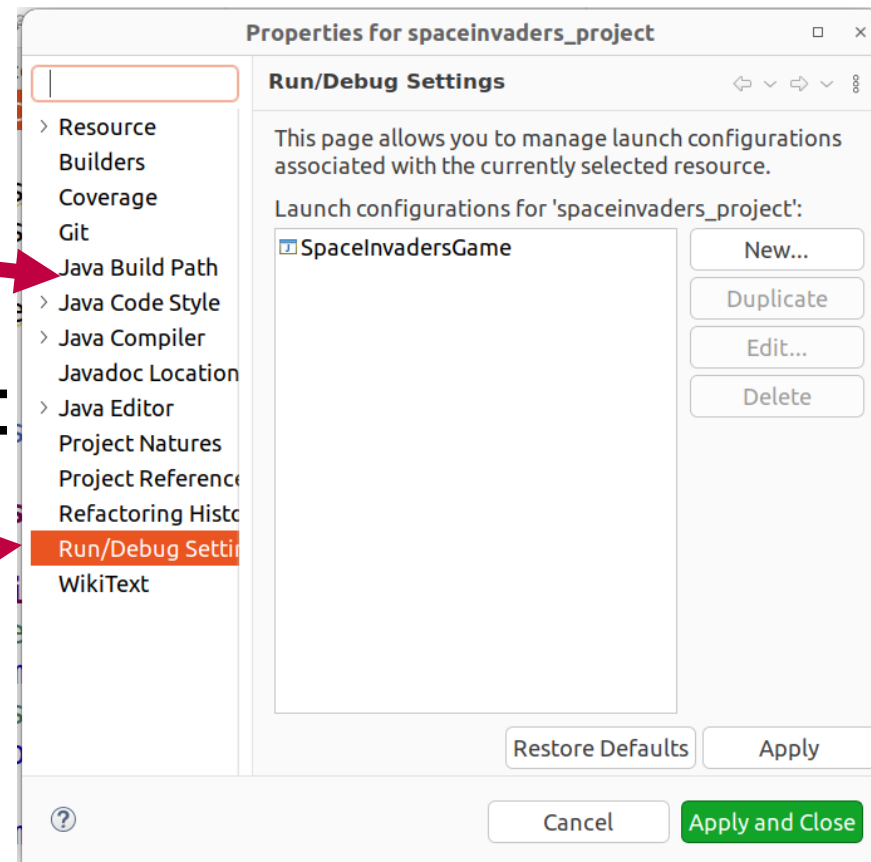
- Projekt auswählen, dann:
Rechtsklick
- „Properties“ wählen
(ganz unten)





Abhängigkeiten eines Projekts verwalten

- CLASSPATH zur Kompilierzeit
- Laufzeit-Einstellungen:
 - ausführbare Klasse
 - Parameter
 - CLASSPATH





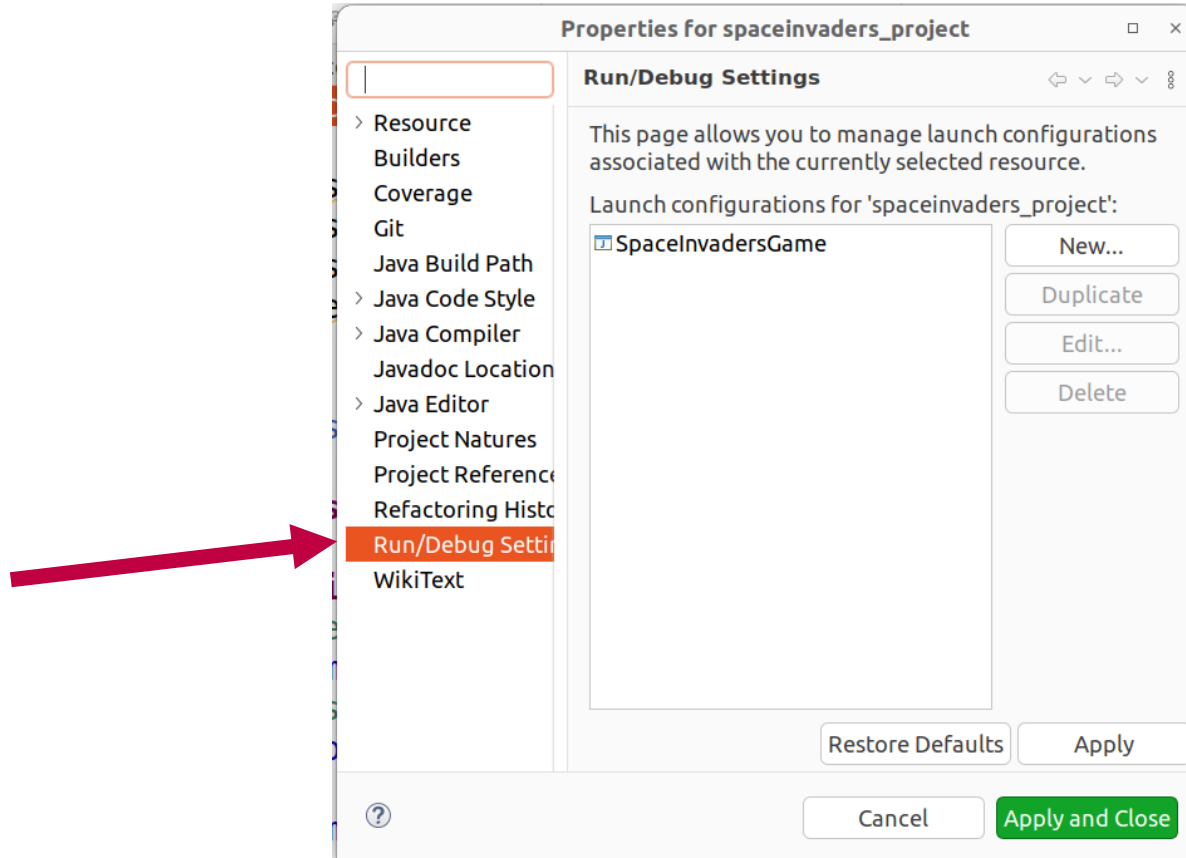


Arbeitsverzeichnis eines Projekts einstellen

- Das Arbeitsverzeichnis („working directory“) gibt ein Verzeichnis an, auf das sich alle relativen Pfadangaben beziehen.
- Falls ein Java-Programm also das File „test.txt“ öffnen will, ist das relativ zum Arbeitsverzeichnis

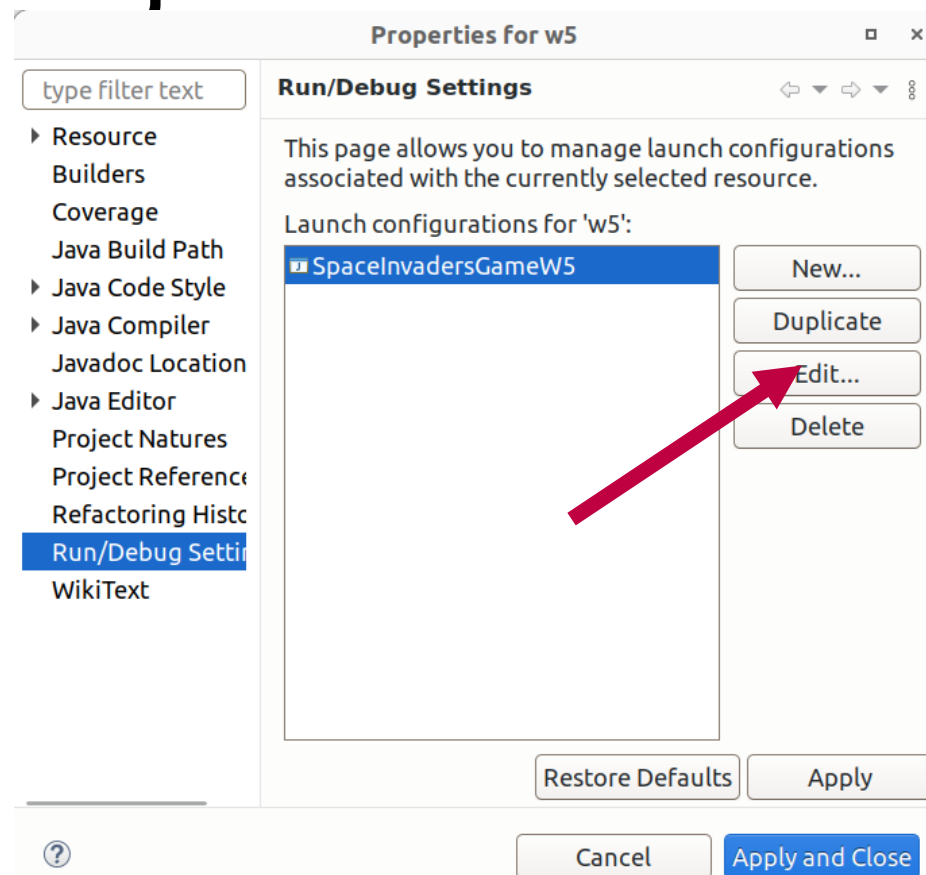


Arbeitsverzeichnis eines Projekts einstellen



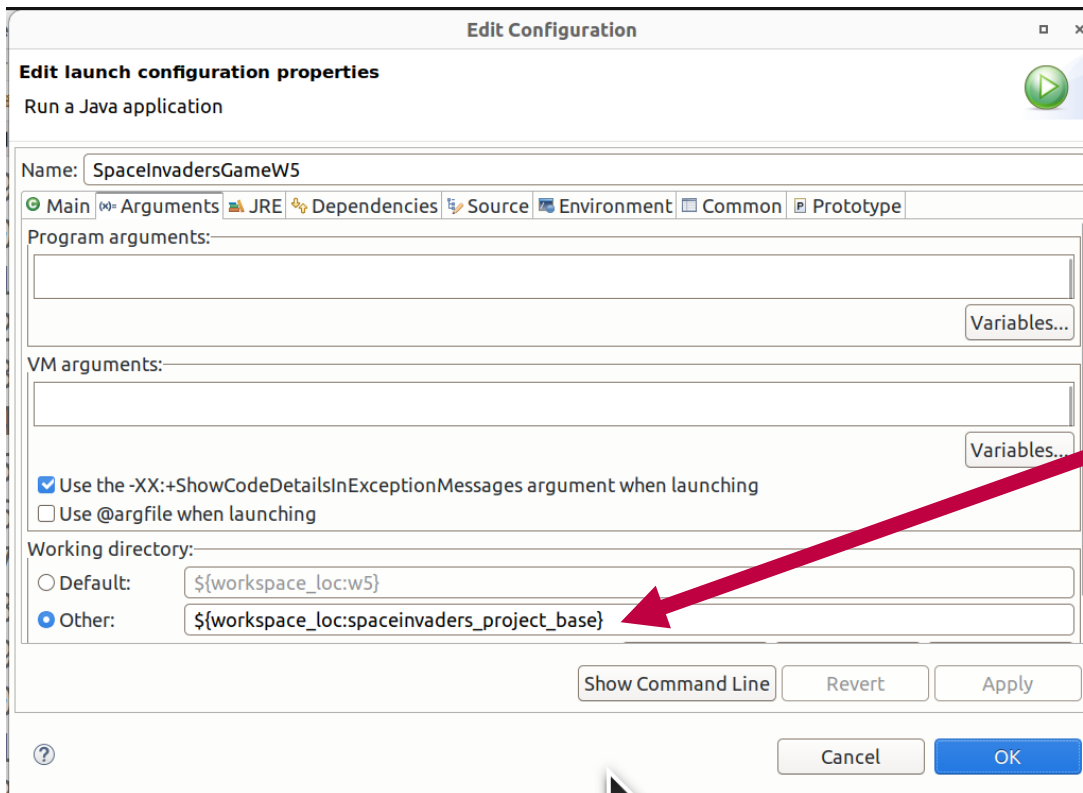


Arbeitsverzeichnis eines Projekts einstellen





Arbeitsverzeichnis eines Projekts einstellen



Anpassen!





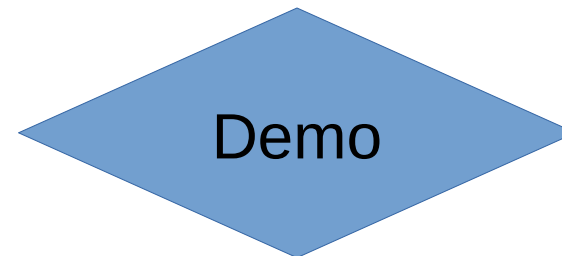
Fortgeschrittene IDE-Funktionen

- Eclipse kann sehr weit reichende Operationen auf Ihrem Quellcode ausführen
 - Formatieren (z.B. Google Java Style)
 - Klassen umbenennen
 - Importe organisieren
 - try/catch einfügen
 - Schleife einfügen
 - Aus/Einkommentieren
 - ...



Fortgeschrittene IDE-Funktionen

- Eclipse erlaubt es Ihnen zu debuggen!
 - das bedeutet: ein Programm Schritt für Schritt auszuführen
 - und nach jedem Schritt alle Variablen/Attribute anzusehen
 - damit kann man komplexe Fehler sehr effizient finden!





CUT: Q&A