

Aufgabe 1 (1 Bonuspunkt)

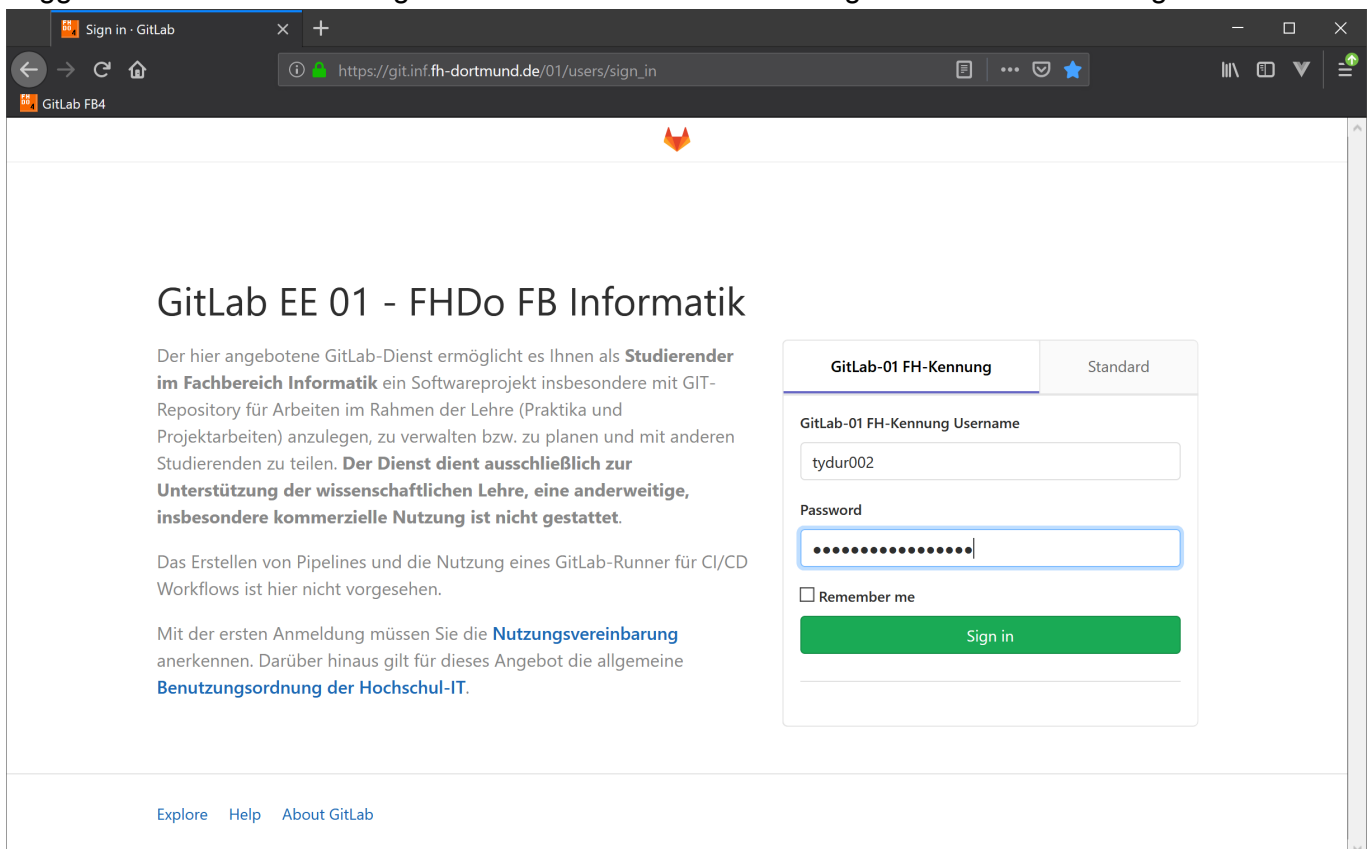
1. Projekt in GitLab und Eclipse einrichten

Ziel dieser Aufgabe ist es, ein Projekt in GitLab zu erstellen und dieses als Java-Projekt in Eclipse einzurichten. Folgen Sie dazu der folgenden Schritt-für-Schritt-Anleitung.

Wichtig: Bitte führen Sie die Schritte **exakt** wie in der folgenden Anleitung beschrieben aus! Schon kleine Abweichungen (z.B. zu früher Klick auf "Finish") können zu einem nicht funktionierenden Zustand führen.

1.1. GitLab

1. Rufen Sie GitLab unter der URL <https://git.inf.fh-dortmund.de> auf.
2. Loggen Sie sich unter der Registerkarte "GitLab-01 FH-Kennung" mit Ihrer FH-Kennung in GitLab ein:



Sign in · GitLab

https://git.inf.fh-dortmund.de/01/users/sign_in

GitLab EE 01 - FHDo FB Informatik

Der hier angebotene GitLab-Dienst ermöglicht es Ihnen als **Studierender im Fachbereich Informatik** ein Softwareprojekt insbesondere mit GIT-Repository für Arbeiten im Rahmen der Lehre (Praktika und Projektarbeiten) anzulegen, zu verwalten bzw. zu planen und mit anderen Studierenden zu teilen. **Der Dienst dient ausschließlich zur Unterstützung der wissenschaftlichen Lehre, eine anderweitige, insbesondere kommerzielle Nutzung ist nicht gestattet.**

Das Erstellen von Pipelines und die Nutzung eines GitLab-Runner für CI/CD Workflows ist hier nicht vorgesehen.

Mit der ersten Anmeldung müssen Sie die [Nutzungsvereinbarung](#) anerkennen. Darüber hinaus gilt für dieses Angebot die allgemeine [Benutzungsordnung der Hochschul-IT](#).

GitLab-01 FH-Kennung Standard

GitLab-01 FH-Kennung Username

tydur002

Password

.....

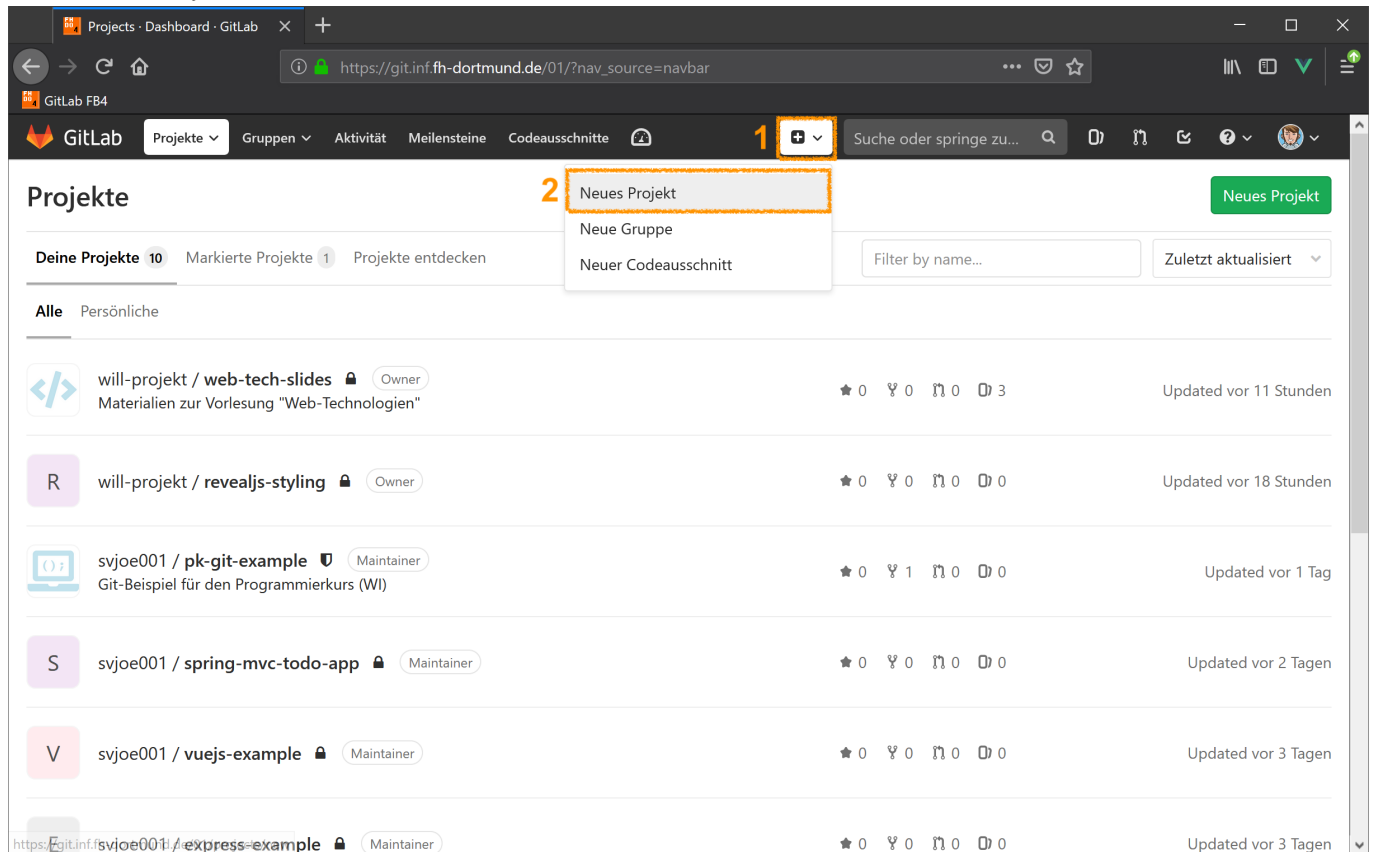
☐ Remember me

Sign in

Explore Help About GitLab

3. Vervollständigen Sie Ihr Profil mit Klarnamen und E-Mail-Adresse (z.B. über das runde Profilbild in der Menüleiste rechts oben → Menüpunkt "Einstellungen" → "Profil"). Dies ist wichtig für die Zuordnung und Vergabe der Bonuspunkte.
4. Wählen Sie in der Menüleiste die Schaltfläche "+" (1) und dort den Menüpunkt "Neues Projekt" (2), um

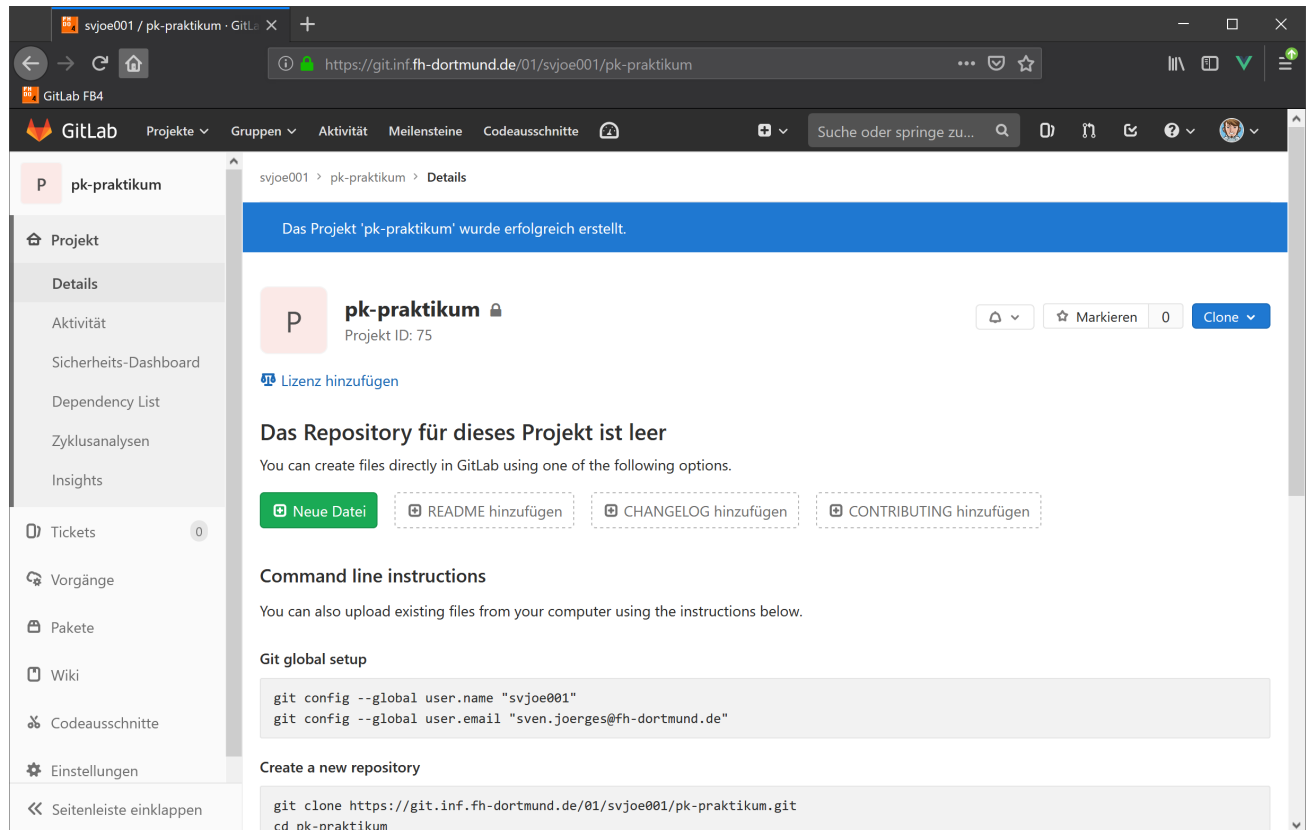
ein neues Projekt zu erstellen:



5. Im folgenden Formular sind vor Allem zwei Einstellungen wichtig:

1. Geben Sie Ihrem Projekt unter "Projektname" (1) den Namen `pk-praktikum`.
2. Stellen Sie die Sichtbarkeit Ihres Projektes unter "Visibility Level" (2) auf "Privat".

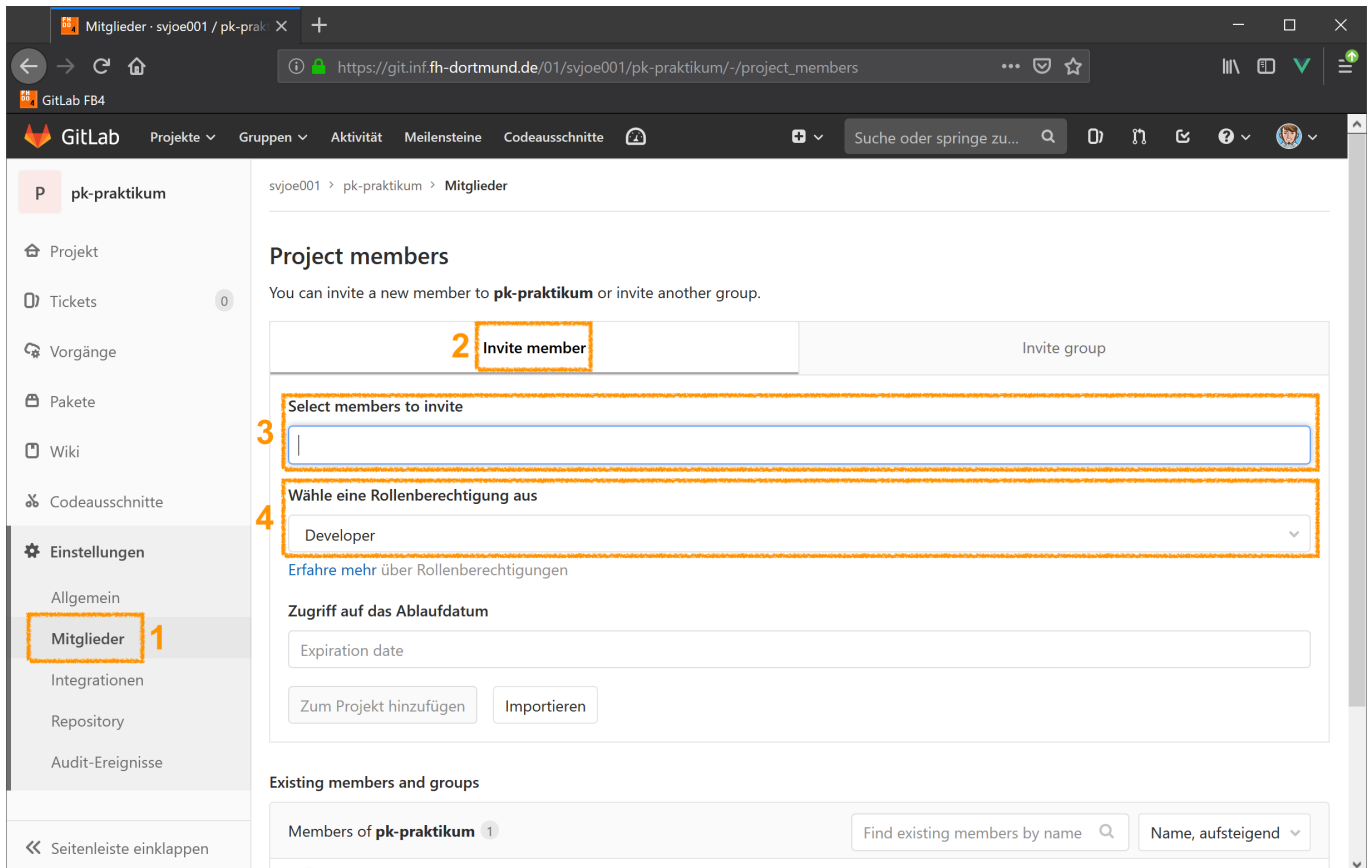
Nach Klick auf "Create project" (3) sieht Ihr neues leeres Projekt wie folgt aus:



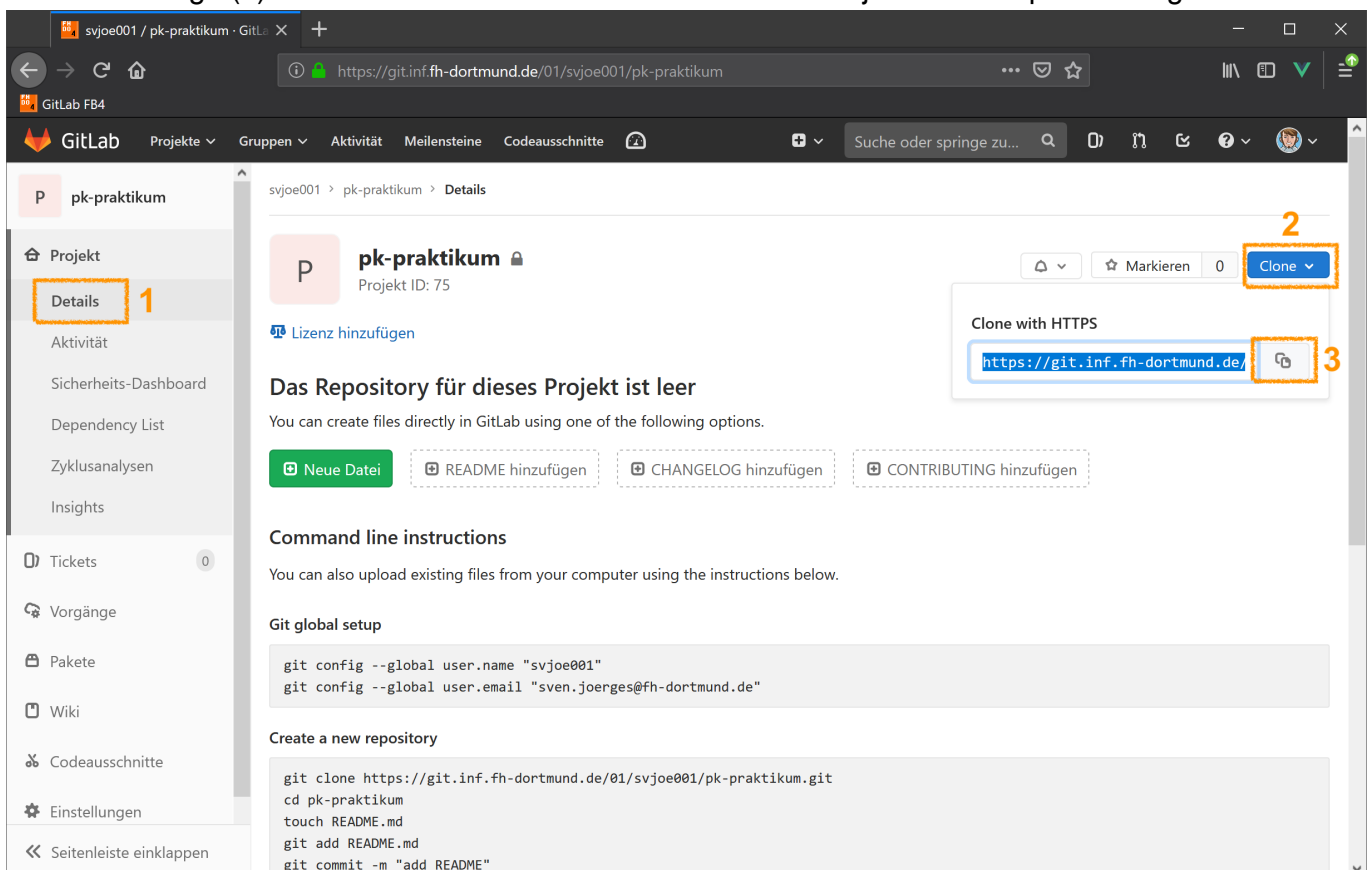
6. Wechseln Sie in Ihrem Projekt nun im Menü links in den Bereich "Einstellungen" → "Mitglieder" (1). Wählen Sie rechts auf der Seite "Project members" die Registerkarte "Invite member" (2). Tragen Sie im Feld "Select members to invite" (3) die folgenden Benutzer ein:

- `svjoe001` Sven Jörges
- `koko1002` Konstantin Koll
- `famur001` Fathi Murad
- `mosch008` Moritz Schäfer
- `legue002` Lenard Güldenpenning

Stellen Sie unter "Wähle eine Rollenberechtigung aus" (4) die Rolle "Developer" ein.



7. Wechseln Sie erneut auf die Detailseite Ihres Projektes: links im Bereich "Projekt" → "Details" (1). Betätigen Sie die Schaltfläche "Clone" (2) und kopieren Sie sich die angezeigte URL in die Zwischenablage (3). Diese URL werden Sie zum Klonen des Projektes in Eclipse benötigen.

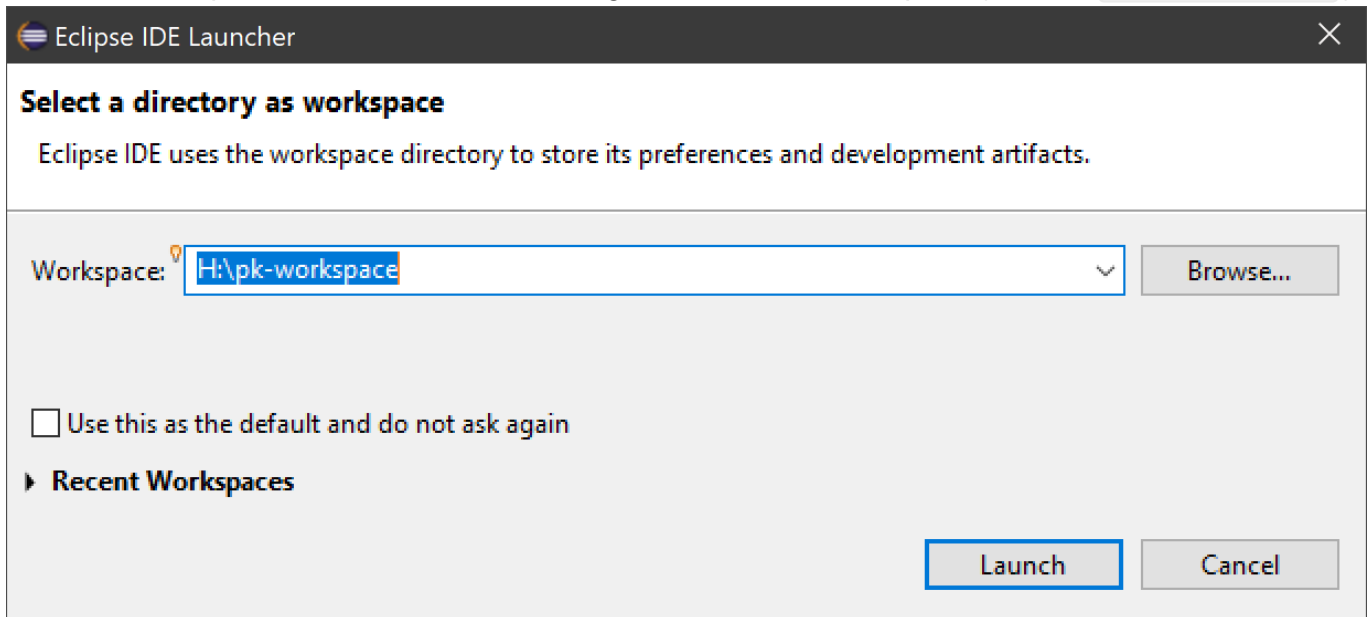


1.2. Eclipse

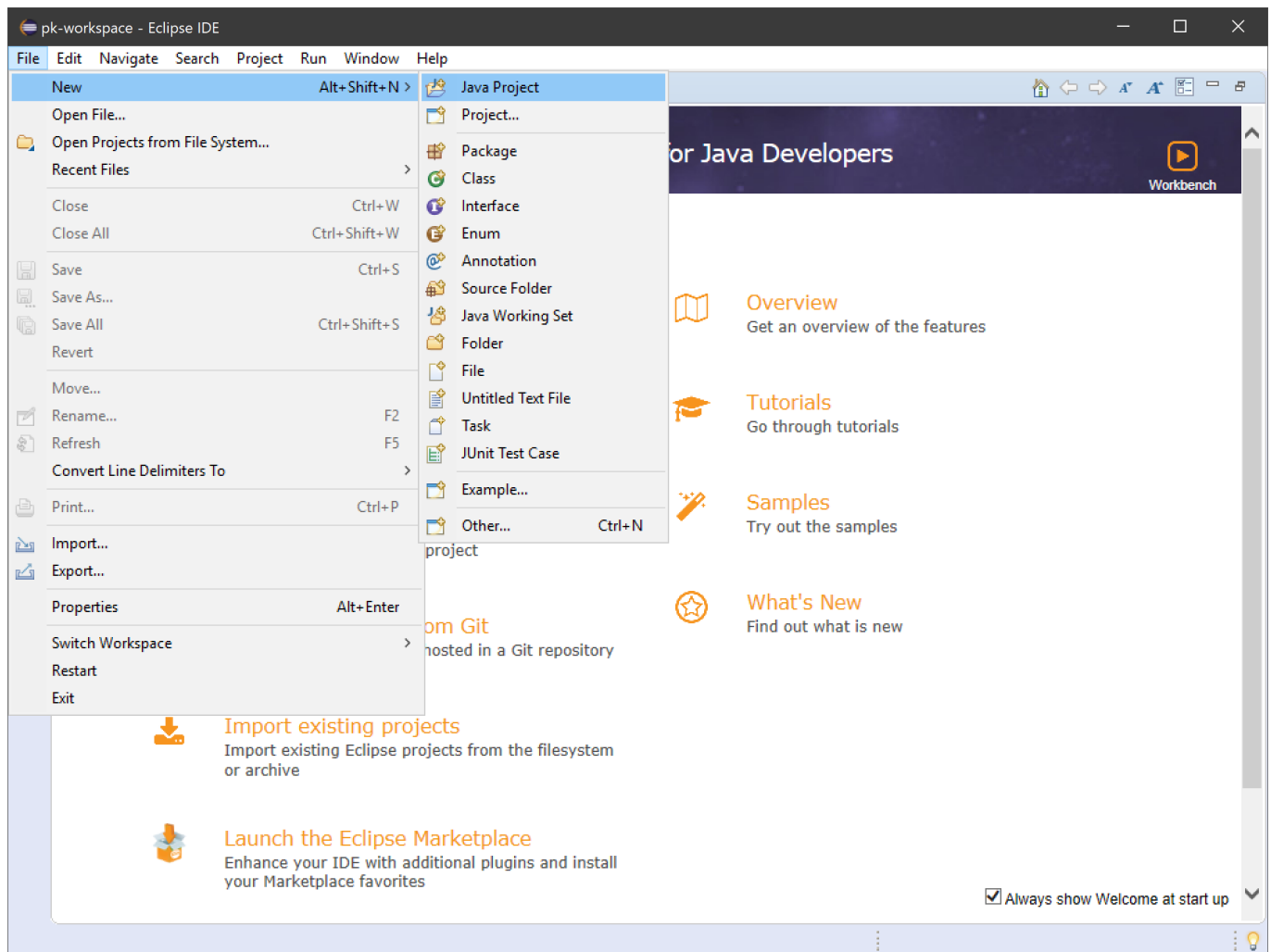
Wichtig: Die folgenden Ausführungen gehen von "Eclipse Java 2020-09" in Kombination mit Java 11 aus. **Auf ältere Versionen von Java bzw. Eclipse ist die folgende Anleitung ggf. nicht anwendbar!** Bitte installieren Sie auf Ihrem Rechner Java 11 sowie ein aktuelles Eclipse:

- Java 11: <https://adoptopenjdk.net/>
- Eclipse: <https://www.eclipse.org/downloads/>

1. Starten Sie Eclipse. Erstellen Sie auf Nachfrage einen neuen Workspace (im Bild: `pk-workspace`):



2. Wählen Sie im Menü "File" → "New" → "Java Project":



3. Es öffnet sich das Fenster "New Java Project". Geben Sie Ihrem Java-Projekt unter "Project name" einen Namen (im Bild: `pk-praktikum`). Die Einstellung "Use default location" sollte aktiviert sein, damit das Java-Projekt in unserem Workspace abgelegt wird. Bestätigen Sie die Einstellungen durch Klick auf "Finish".

New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location

Location: [Browse...](#)

JRE

☒ Use an execution environment JRE:

☐ Use a project specific JRE:

☐ Use default JRE (currently 'jdk-11.0.4.11-hotspot') [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

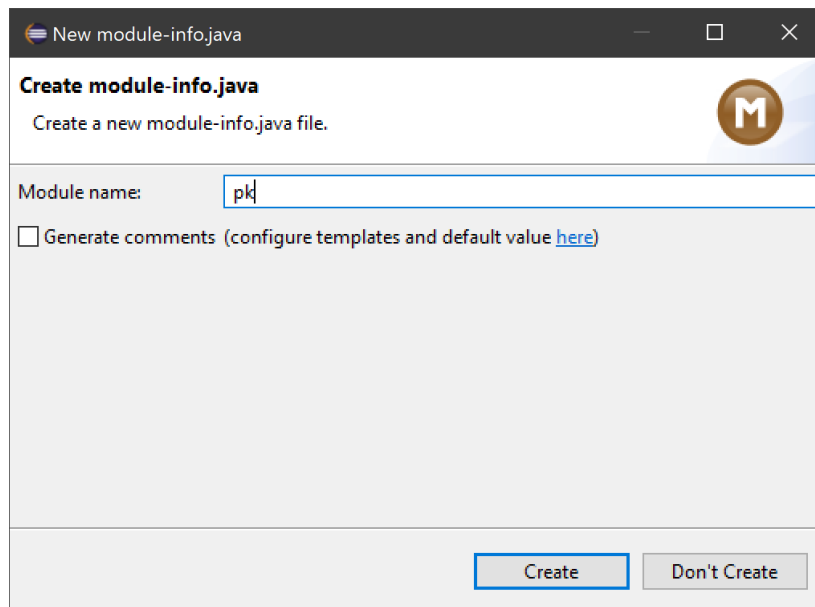
Working sets

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

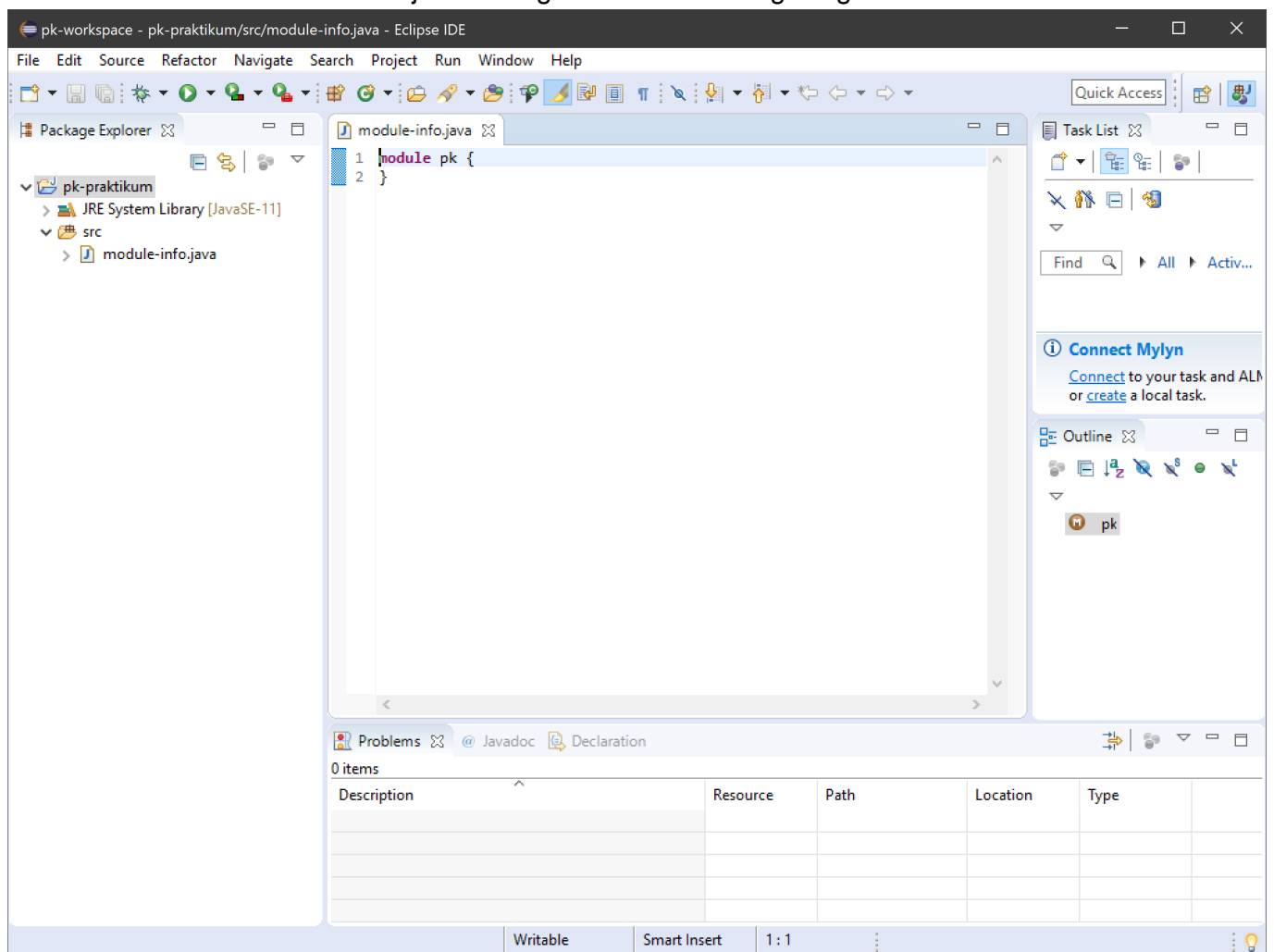
Working sets: [Select...](#)

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

4. Im folgenden Fenster werden Sie gebeten, ein neues Java-Modul anzulegen. Geben Sie unter "Module name" den Namen ein und bestätigen Sie die Eingaben durch Klick auf "Create":

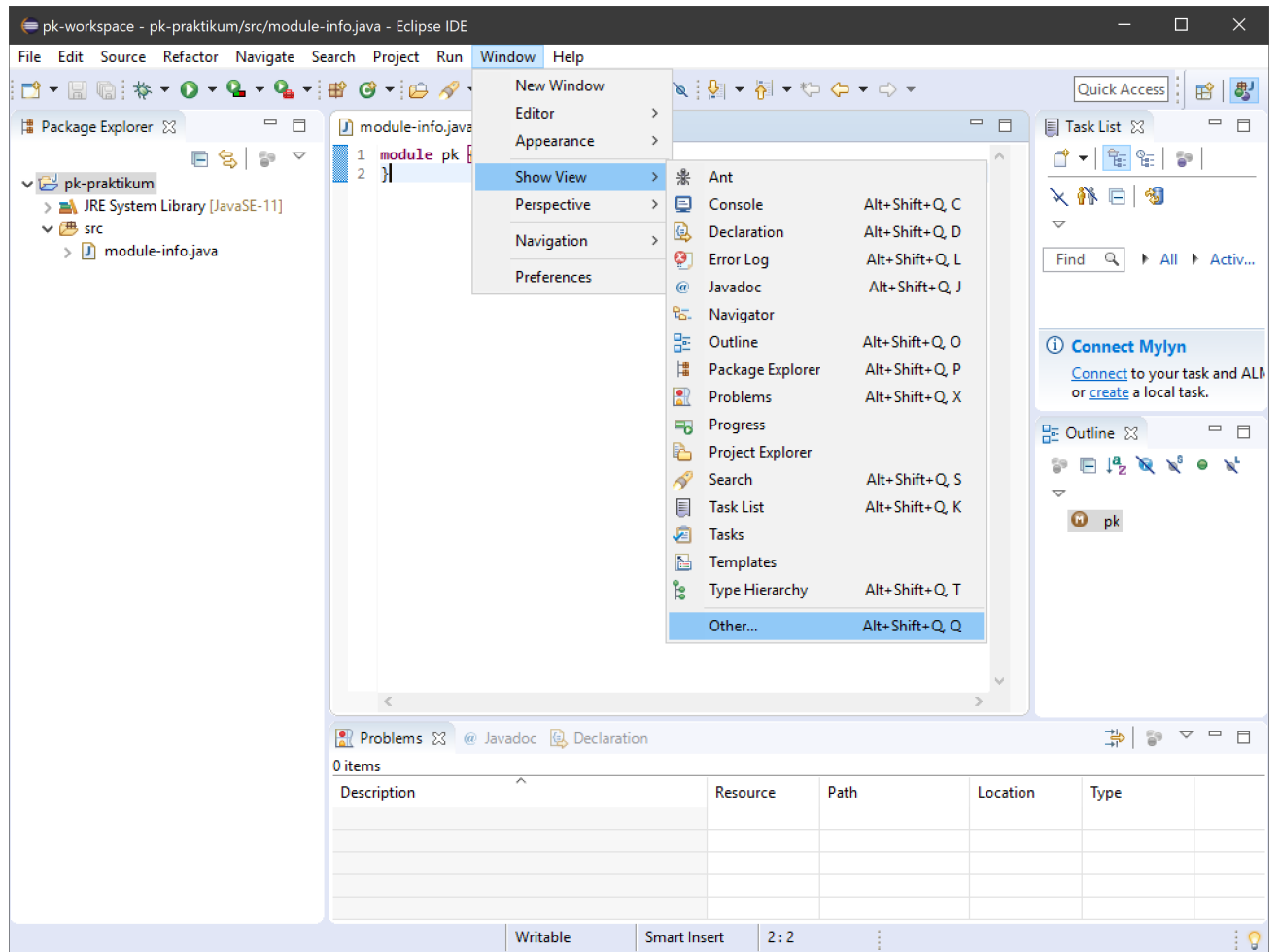


5. Es sollte nun ein neues Java-Projekt mit folgendem Aufbau angezeigt werden:

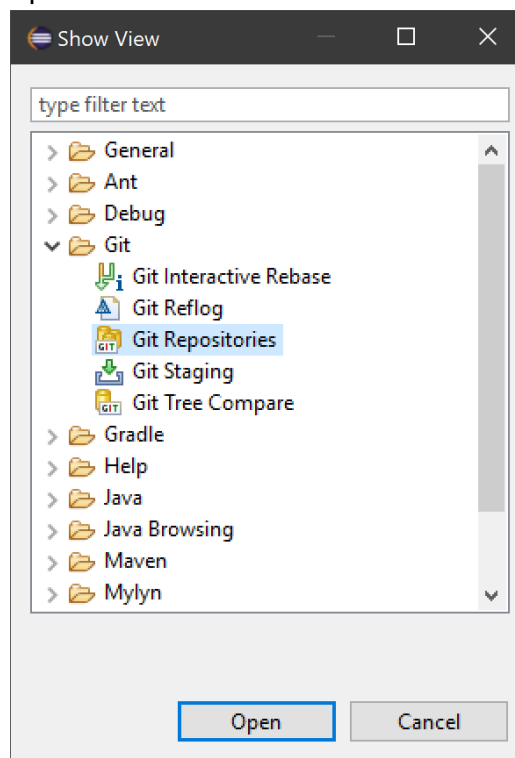


6. Blenden Sie nun den "Git Repositories View" ein:

1. Wählen Sie im Menü "Window" → "Show View" → "Other...":

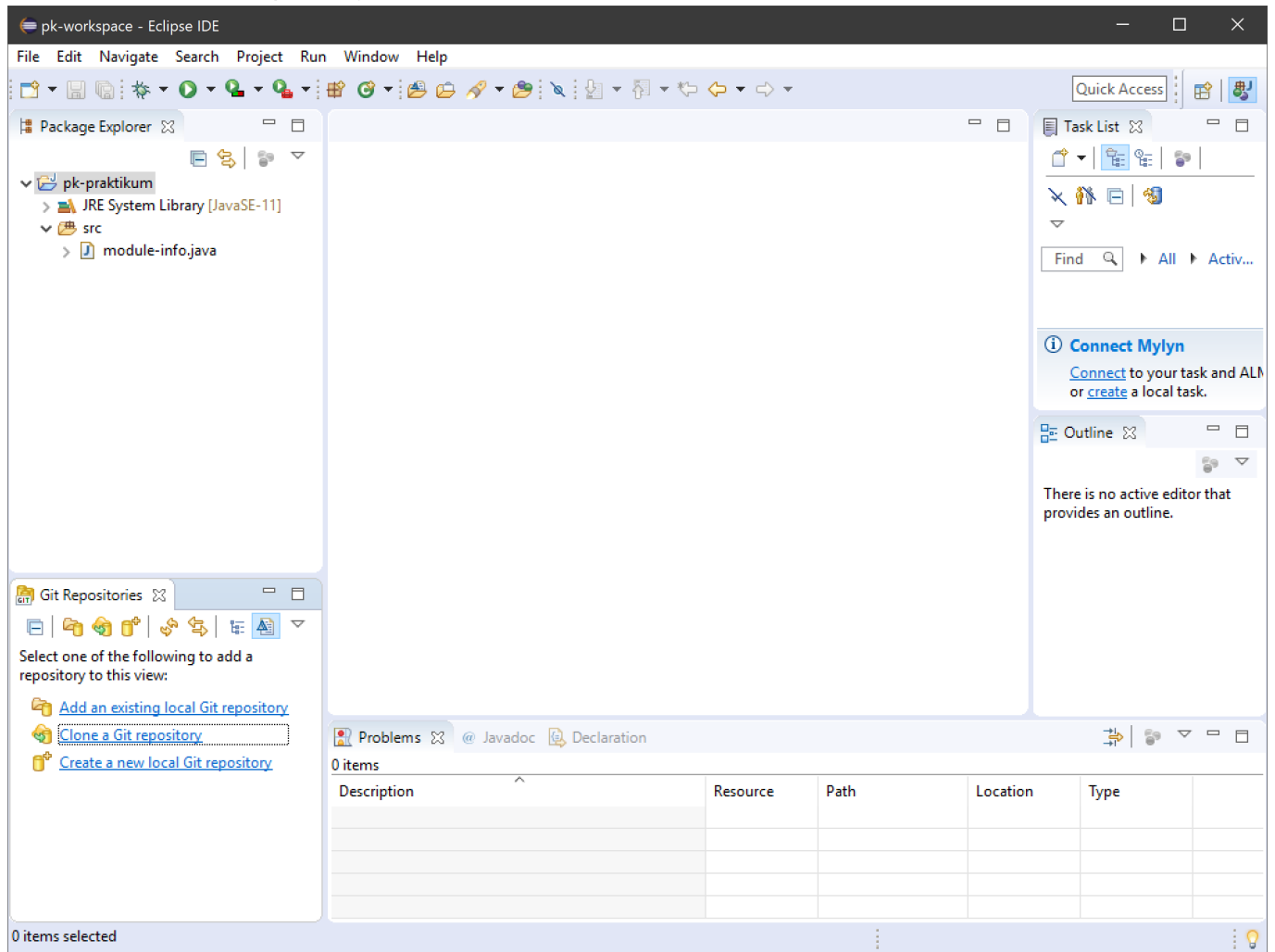


2. Wählen Sie im sich öffnenden Dialog den Eintrag "Git" → "Git Repositories" und bestätigen Sie die Auswahl durch Klick auf "Open":



7. Der "Git Repositories View" sollte nun in Eclipse erscheinen (im Bild: links unten). Wählen Sie dort die

Aktion "Clone a Git repository" aus:



8. Tragen Sie im folgenden Fenster unter "URI" die von GitLab kopierte URL Ihres Projektes ein. Die Felder "Host" und "Repository path" sollten sich automatisch füllen. Tragen Sie unter "Authentication" Ihre FH-Kennung ein. Falls gewünscht, können Sie optional die Einstellung "Store in Secure Store" aktivieren, damit Eclipse sich Ihr Passwort für die Zukunft merkt. Klicken Sie "Next >".

Clone Git Repository

Source Git Repository

Enter the location of the source repository.

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:


Password:

☒ Store in Secure Store

9. Eclipse sollte nun ein leeres Repository erkennen. Klicken Sie "Next >".

Clone Git Repository

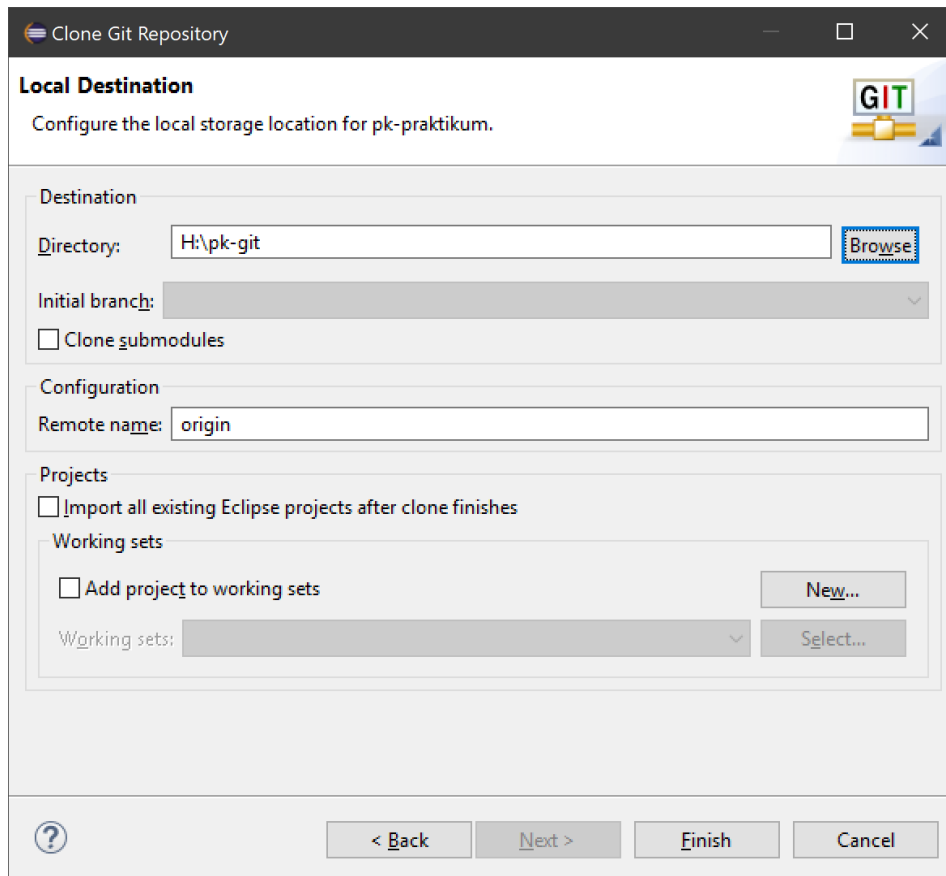
Branch Selection

 Source Git repository is empty

Branches of https://git.inf.fh-dortmund.de/01/svjoe001/pk-praktikum.git:

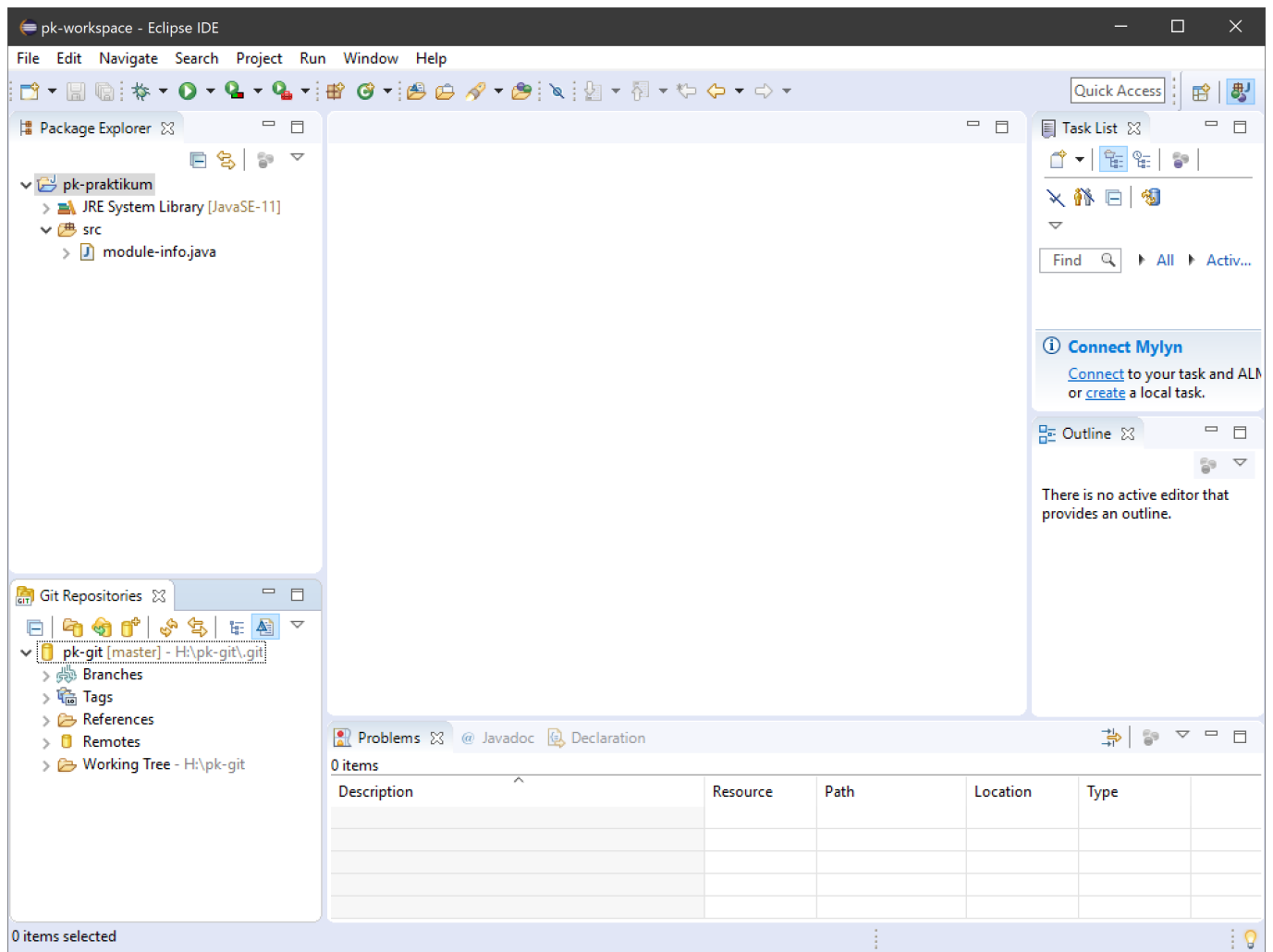
10. Geben Sie unter "Directory" ein Verzeichnis an, in welches Ihr Git-Repository geklont werden soll (im Bild: `H:\pk-git`).

Wichtig: Legen Sie hier unbedingt ein separates Verzeichnis an, welches **nicht** im Eclipse-Workspace oder im oben genannten Java-Projekt liegt. Eclipse fordert eine strikte Trennung des *lokalen Repositories* und der *Arbeitskopie*.

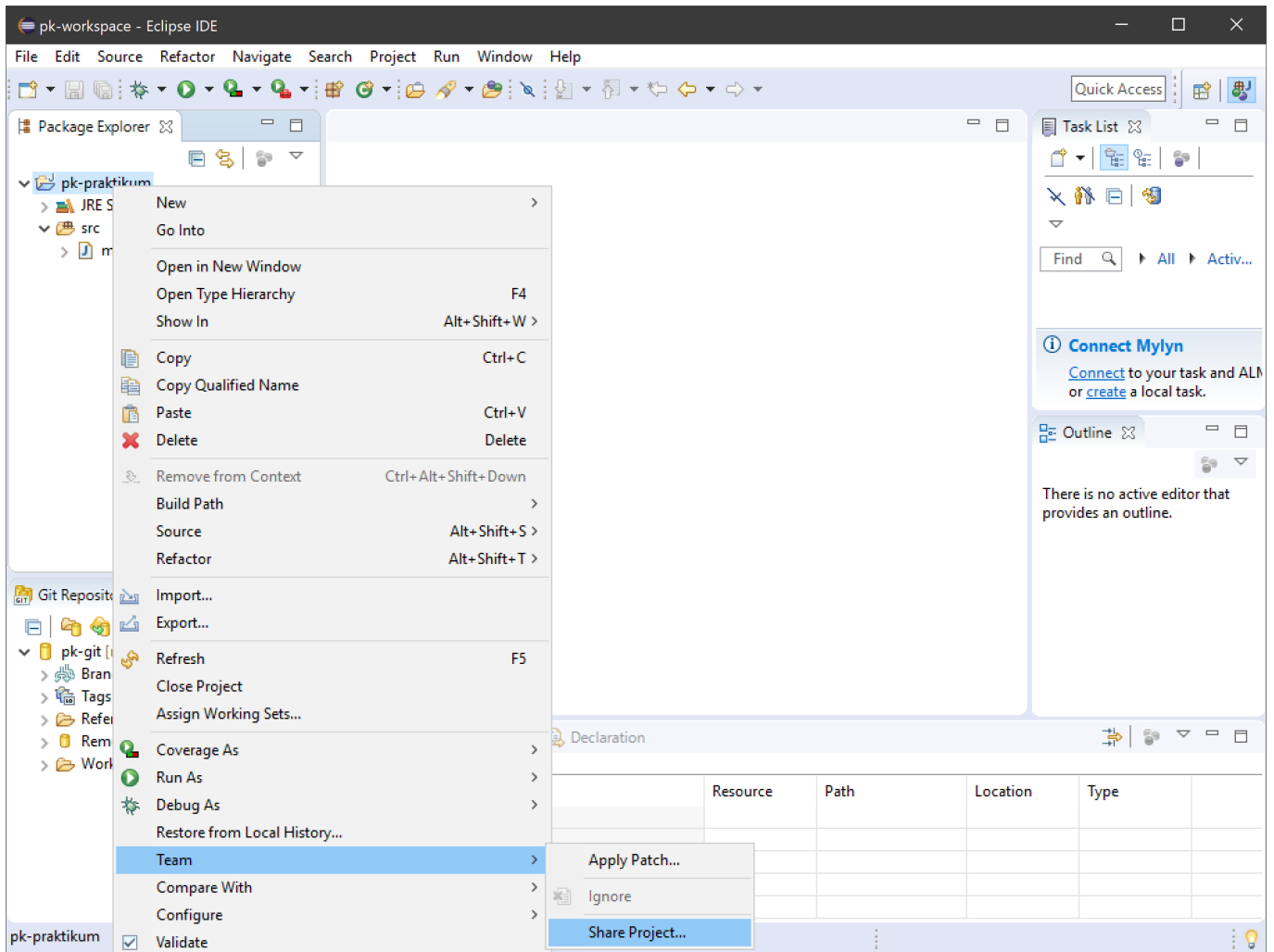


Bestätigen Sie die Eingaben durch Klick auf "Finish".

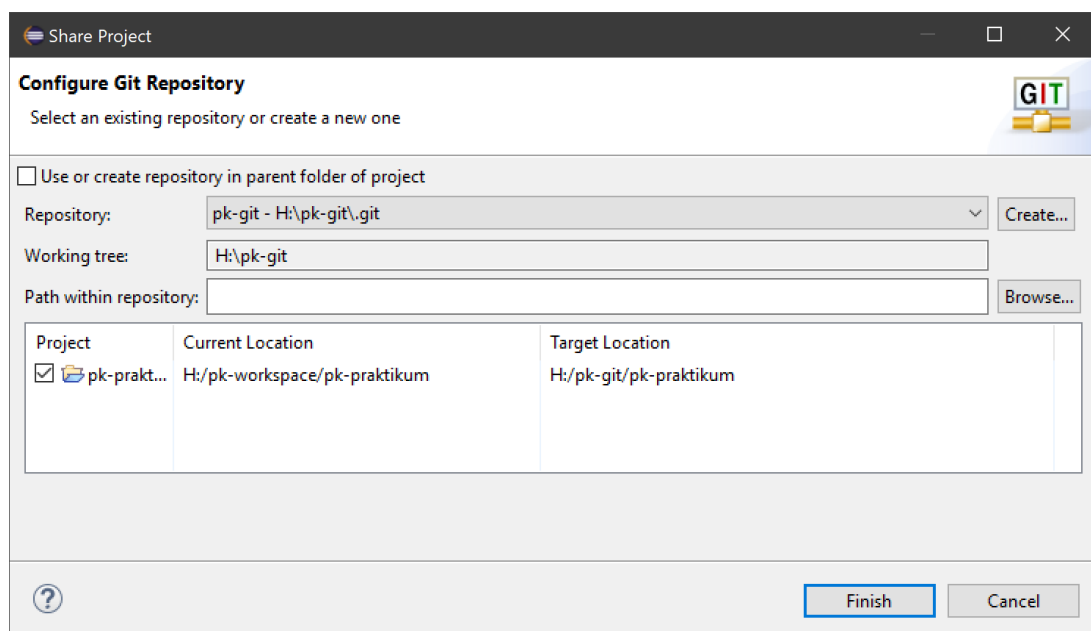
11. Im "Git Repositories View" sollte nun ein Repository wie folgt angezeigt werden:



- Nun müssen wir noch unser Java-Projekt mit dem Repository verbinden. Wählen Sie im Kontextmenü des Java-Projektes den Punkt "Team" → "Share Project...":



13. Wählen Sie im folgenden Fenster unter "Repository" unser gerade geklontes Repository aus (im Bild: `pk-git`):



Sie haben nun ein Java-Projekt in Eclipse, das Ihr GitLab-Projekt als *Remote Repository* "origin" verwendet.

2. Erstes Java-Programm und Git

1. Legen Sie im Java-Projekt nun ein neues Package `pk` sowie darin eine neue Klasse `FirstProgramm` an.

2. Implementieren Sie in dieser Klasse die Methode

`maleTreppe(int hoehe, int stufentiefe)`. Diese soll eine Treppe aus `hoehe` Stufen der Tiefe `stufentiefe` durch Ausgabe von `*` auf der Konsole zeichnen. Beispielausgabe für `hoehe` =8 und `stufentiefe` =3:

```
***
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

Über die Parameter `hoehe` und `stufentiefe` soll die Treppenform angepasst werden können.

3. Ergänzen Sie eine `main`-Methode, welche die Methode `maleTreppe` mit `hoehe` =6 und `stufentiefe` =2 aufruft.

4. Committen Sie nun den Stand Ihres Projektes in Ihr lokales Repository (Kontextmenü "Team" → "Commit...", es öffnet sich der "Git Staging"-View). Gehen Sie dabei folgendermaßen vor:

1. Betrachten Sie den Bereich "Unstaged Changes": Welche der Dateien in Ihrem Projekt sollten nicht committed werden? Fügen Sie diese Dateien in `.gitignore` hinzu (Rechtsklick auf die Datei → "Ignore").

Hinweis: Die Datei `.gitignore` selbst sowie die Datei `module-info.java` sollten *auf jeden Fall* committed werden!

2. Fügen Sie alle Dateien, die Teil des Commits sein sollen, in der Staging Area ("Staged Changes") hinzu (Benutzung der `+`-Buttons oder per Rechtsklick auf die entsprechende Datei → "Add to Index").

3. Tragen Sie eine sinnvolle Commit-Message ein.

4. Führen Sie den Commit durch (*nicht* "Commit and Push...!").

5. Modifizieren Sie `maleTreppe` nun so, dass die Treppe "anders herum" ausgegeben wird, z.B. für `hoehe` =8 und `stufentiefe` =3:

```
***
*****
*****
*****
*****
*****
*****
*****
*****
```

6. Comitten Sie die Änderung in Ihr lokales Repository. Geben Sie wieder eine sinnvolle Commit-Message an.
7. Übertragen Sie Ihren lokalen Stand an GitLab (entweder durch Verwendung von "Commit and Push..." im "Git Staging" View oder über das Kontextmenü "Team" → "Push Branch 'master'...").
8. Öffnen Sie Ihr Projekt in GitLab und überprüfen Sie, ob alles angekommen ist (z.B. durch Blick in den "Verlauf").
9. Navigieren Sie in GitLab zu der Datei Ihres Java-Programms und klicken Sie darauf, um sich die Inhalte anzuschauen. Wählen Sie nun "Bearbeiten", um die Datei in GitLab zu editieren. Fügen Sie einen Kommentar zu der Klasse hinzu. Geben Sie dann unten eine Commit-Message ein und klicken Sie dann auf "Commit changes".
10. Sie haben nun eine Änderung in GitLab, die Sie in Ihrem Eclipse-Projekt noch nicht haben. Was müssen Sie tun, um in Ihrem Eclipse-Projekt den aktuellen Stand von GitLab zu bekommen? Führen Sie die entsprechende Aktion durch.

3. Bonuspunkt abholen

Zeigen Sie Ihre Lösung (Eclipse-Projekt + Projekt in GitLab) einem der Betreuer im Praktikum, um den Bonuspunkt zu erhalten.