

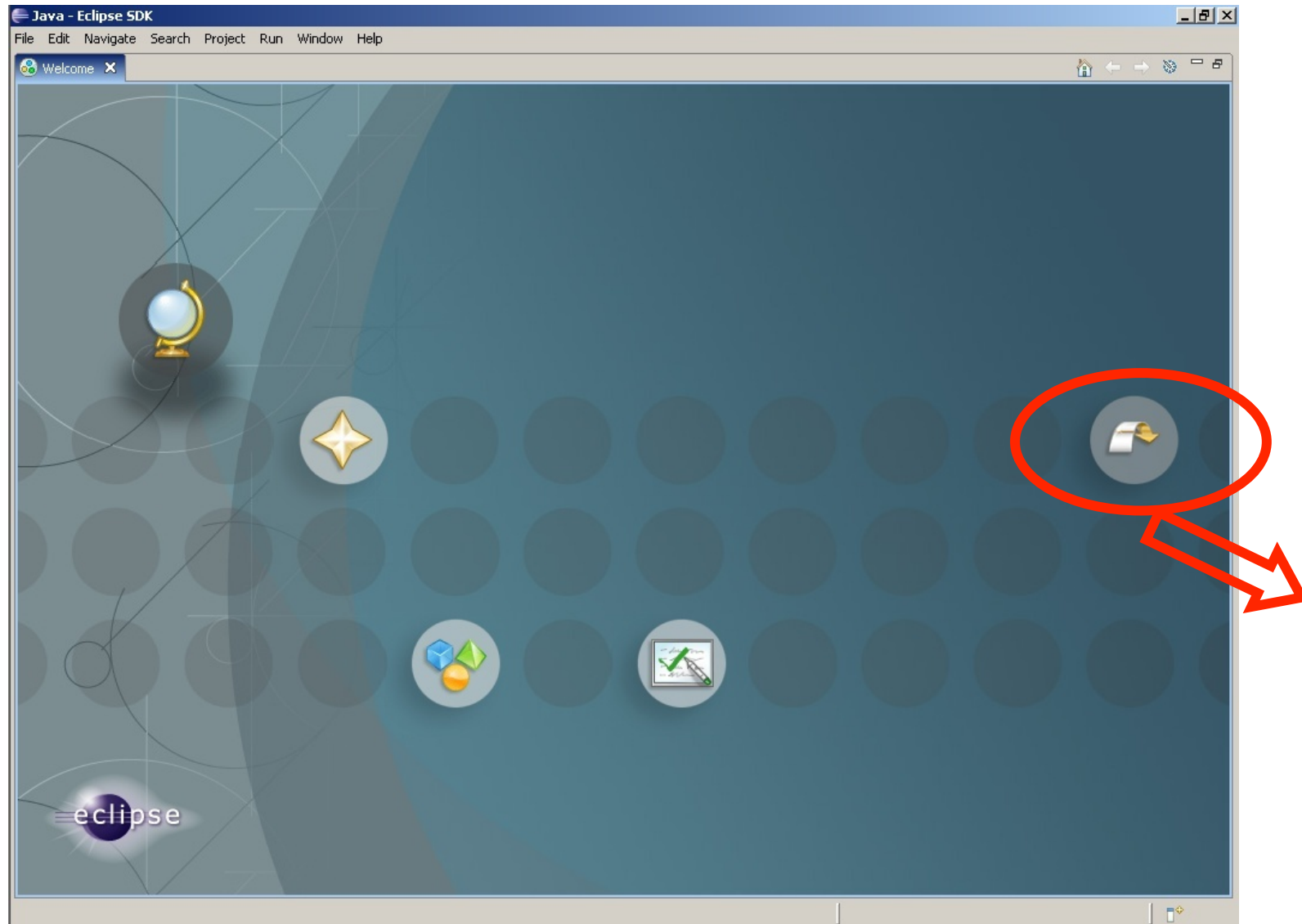
- Hintergrund, Installation, Start von Eclipse
- Arbeiten mit Eclipse
 - Arbeiten mit Git
 - Pakete anlegen
 - Programme / Klassen anlegen
 - Programme ausführen
- Fehlersuche
 - Übersetzungsfehler
 - Laufzeitfehler

- Eclipse ist (unter anderem) Basis für eine Java-Entwicklungsumgebung
- Eclipse wird nach IBM-Anschubfinanzierung als Open Source entwickelt
- Der Quellcode und Binärdateien für verschiedene Plattformen sind unter den Bedingungen der Eclipse Public License (EPL) verfügbar

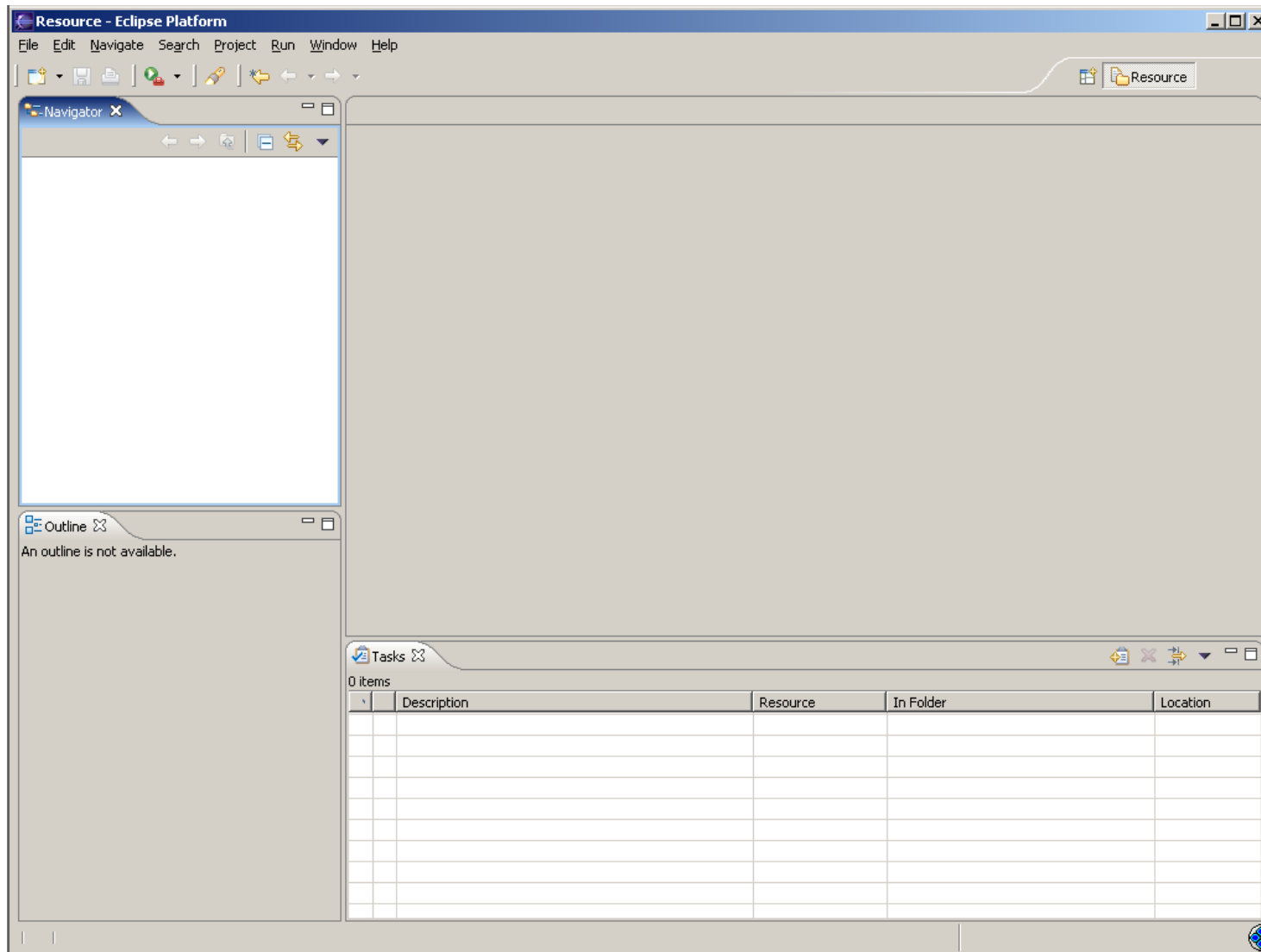
Download und Installation von Eclipse

- Übersetzen etc. der Quellen ist nicht notwendig; für alle unterstützten Plattformen sind Binaries unter www.eclipse.org verfügbar
- Im CIT (Campus IT, Rechenzentrum) ist aktuell die Version „Mars“ installiert, d.h. 4.5 bzw. 4.5.2:
-> Wenn Sie zuhause arbeiten möchten, sollten Sie die gleiche (keine neuere) Version installieren!
- Im CIT ist Java 8 (Versionsnummer 1.8) installiert:
-> Wenn Sie zuhause arbeiten möchten, sollten Sie die gleiche (keine neuere) Version installieren!

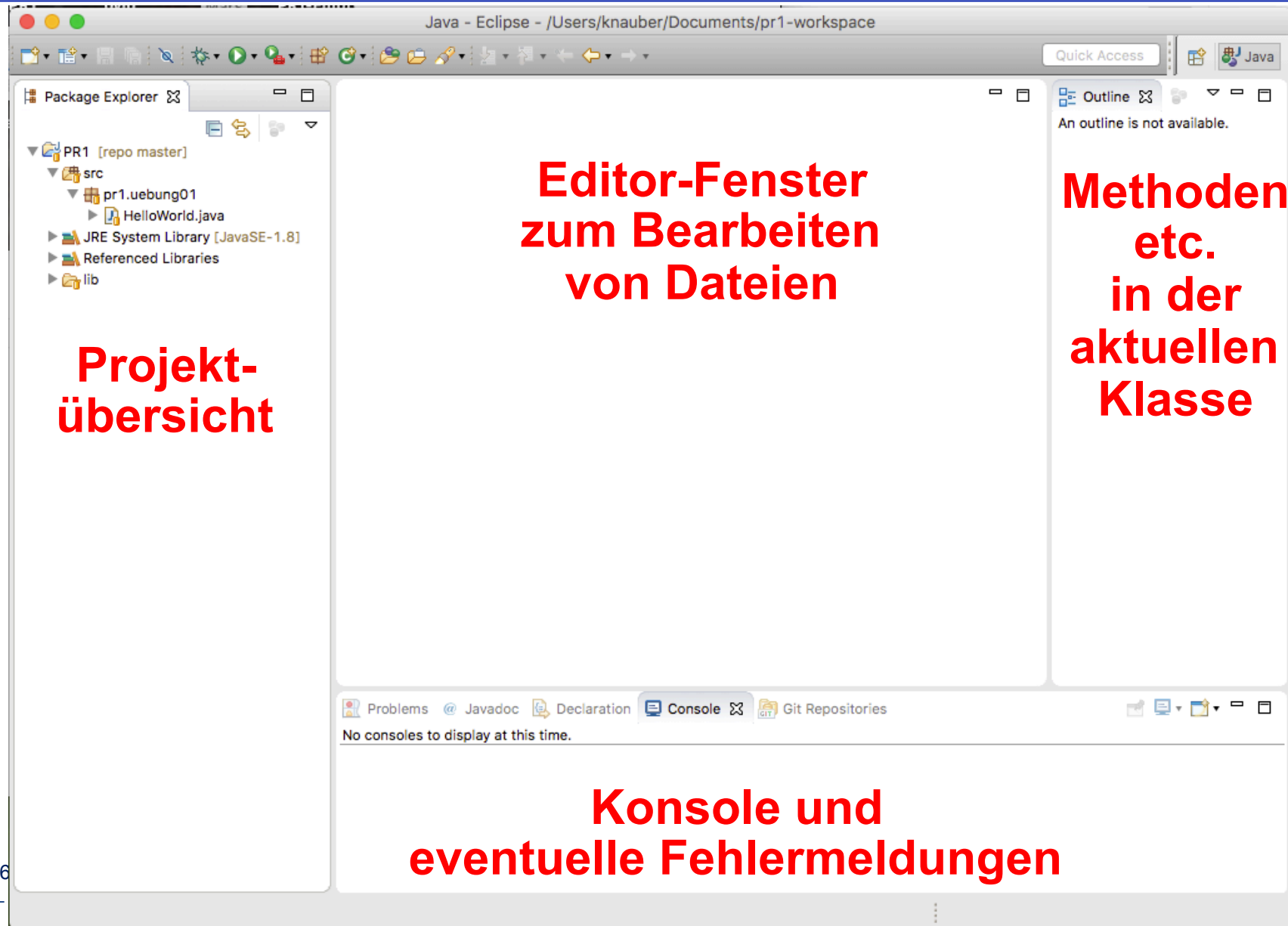
Eclipse starten 1/2



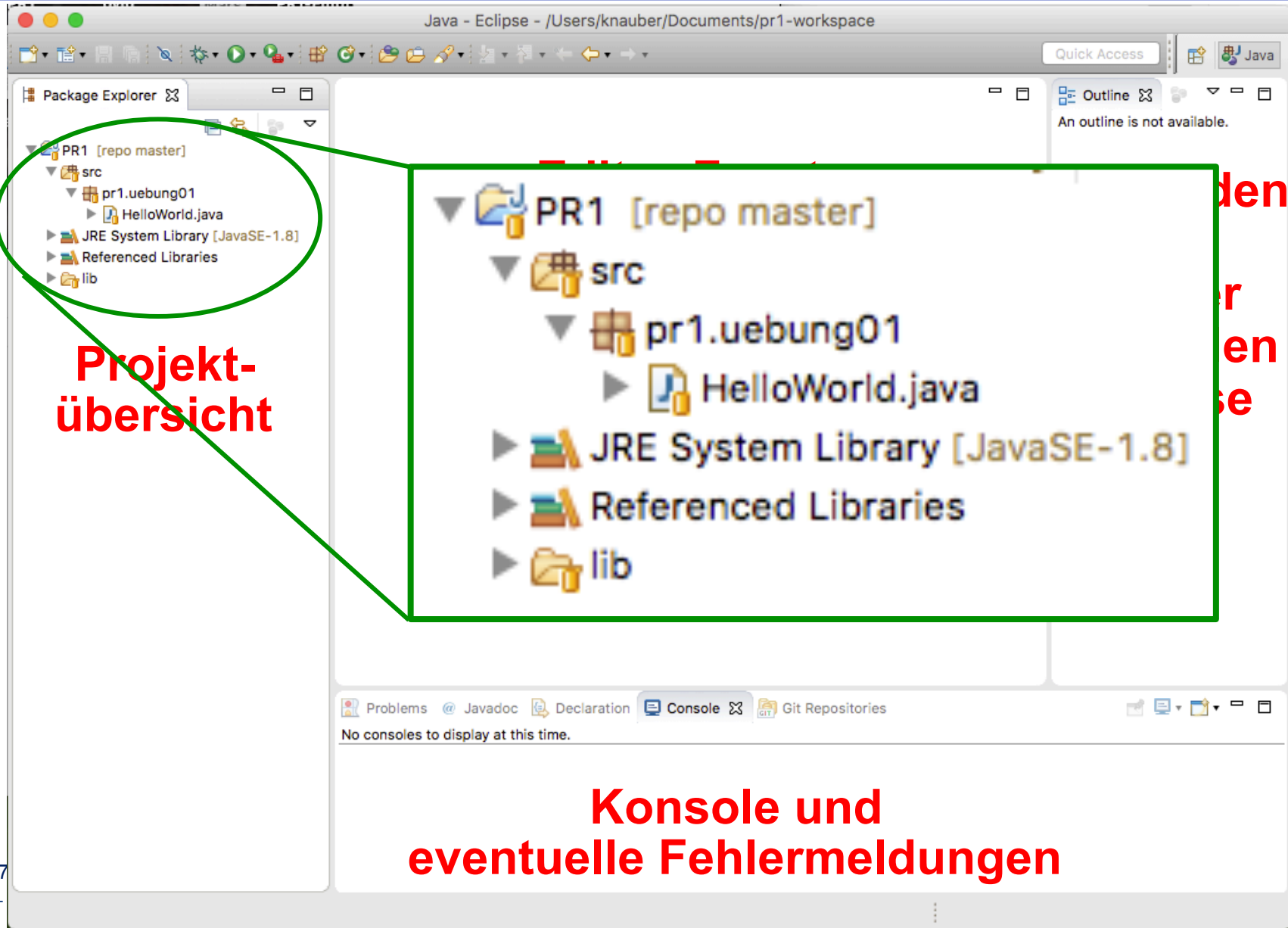
Eclipse starten 2/2



Die „Java-Perspektive“

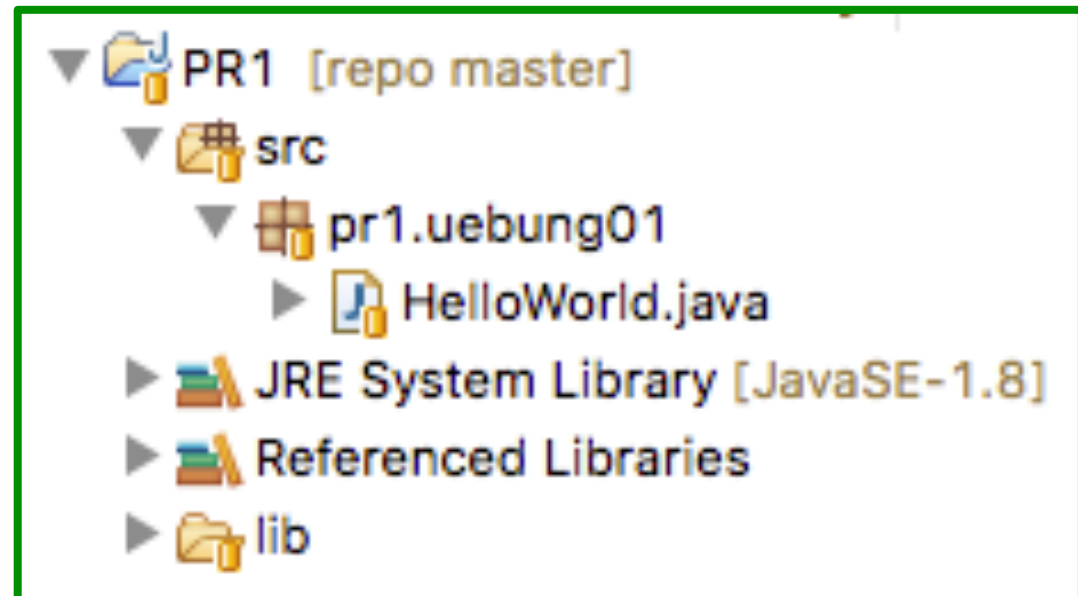


Die „Java-Perspektive“



Organisation in Eclipse

- Alles in Eclipse, was zusammen gehört, ist in sogenannten „Projekten“ organisiert
-> Nennen Sie Ihr Projekt (PR1) nicht um!
- Innerhalb eines Java-Projekts finden wir ein Verzeichnis „src“, in dem unsere Java-Quelltexte (engl. source = Quelle) ablegen
- Wir organisieren sie in „Paketen“ (engl. package), ein Paket pro Übungsblatt



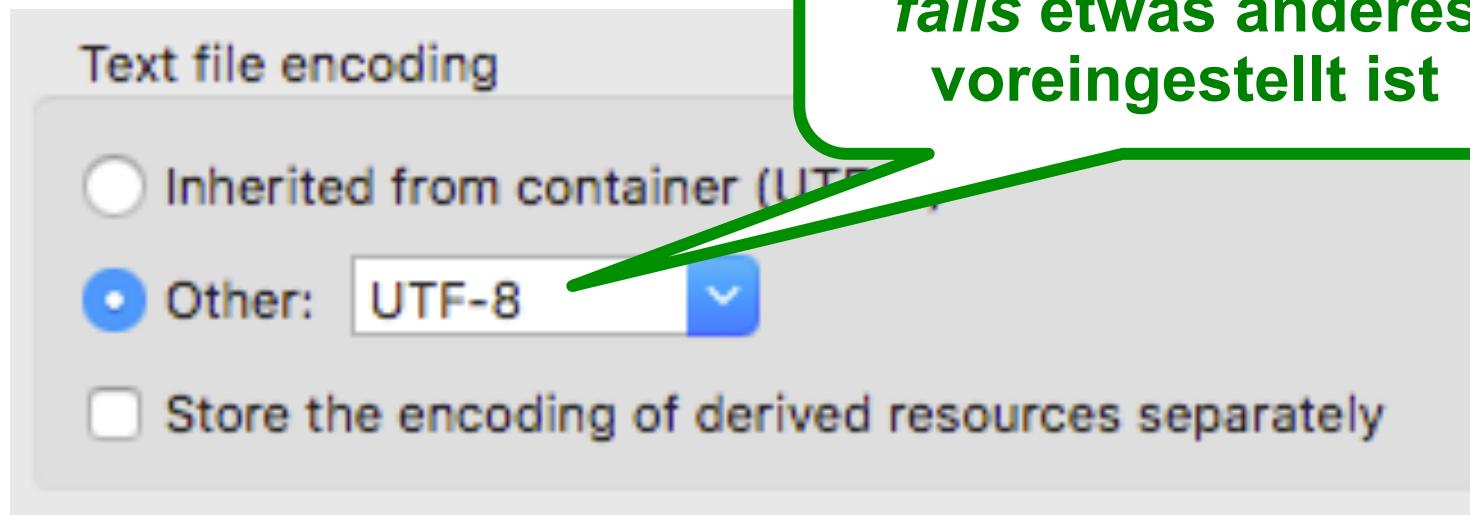
Zeichensatz auf Unicode umstellen

(Verträglichkeit Windows <-> Andere)

Für unser PR1-Projekt benutzen wir die *Unicode-Zeichenkodierung*

- Windows:
Window -> Preferences -> General -> Workspace
- MacOS:
Eclipse -> Preferences -> General -> Workspace

**UTF-8 einstellen,
falls etwas anderes
voreingestellt ist**



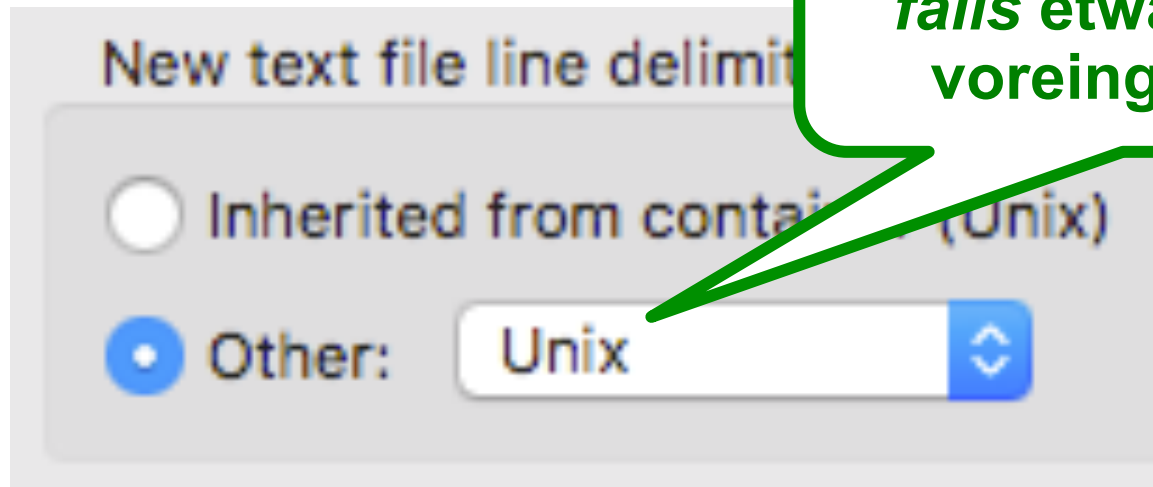
Zeilenumbrüche auf Unix umstellen

(Verträglichkeit Windows <-> Andere)

Für unser PR1-Projekt benutzen wir die *Zeilenumbrüche von Unix*

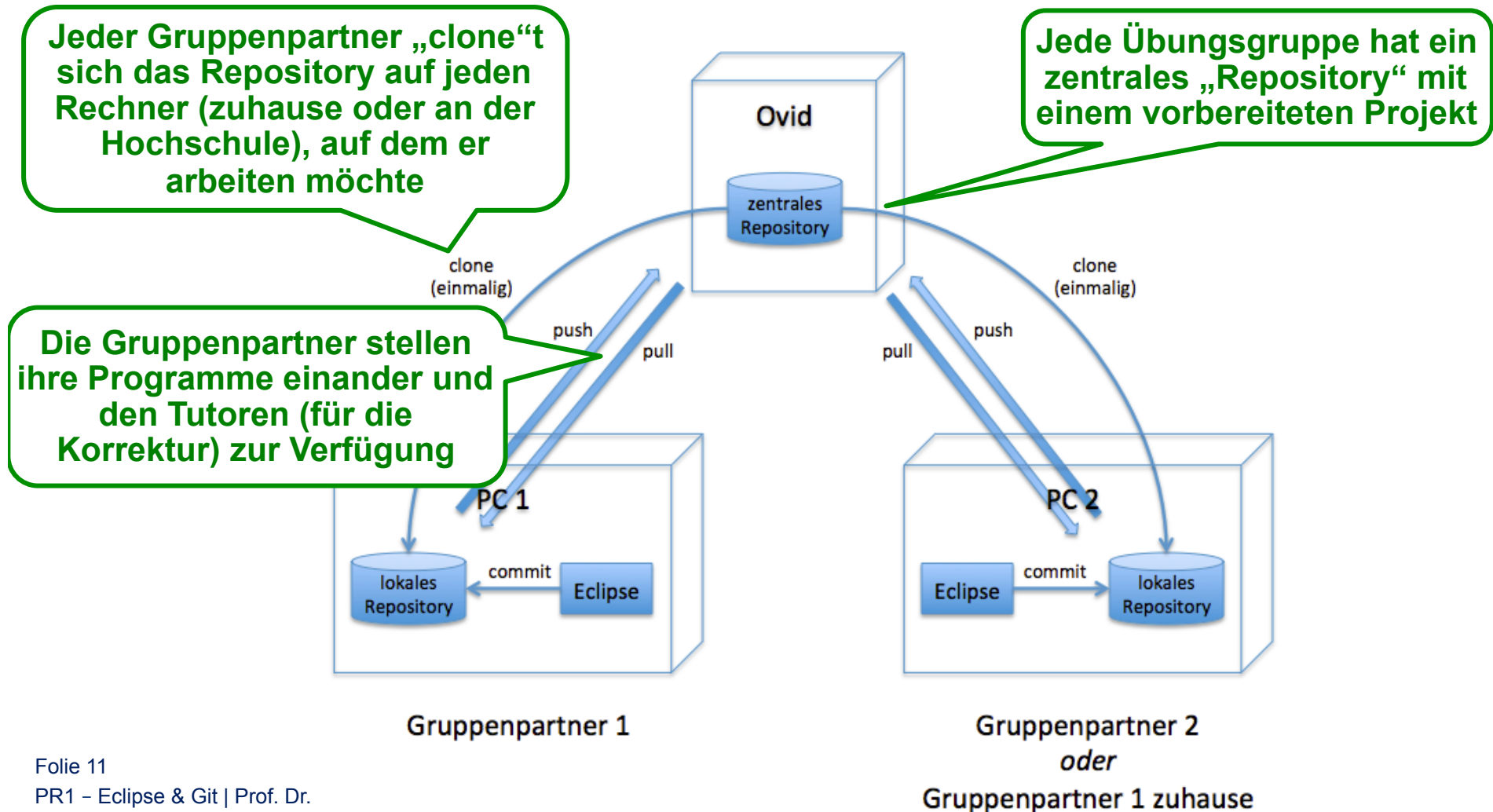
- Windows:
Window -> Preferences -> General -> Workspace
- MacOS:
Eclipse -> Preferences -> General -> Workspace

**Unix einstellen,
falls etwas anderes
voreingestellt ist**



Organisation außerhalb von Eclipse

- Für die Abgabe Ihrer Übungen benutzen Sie das Tool „Git“ (<https://www.atlassian.com/git/>); Prinzip:



Normaler Arbeitsablauf

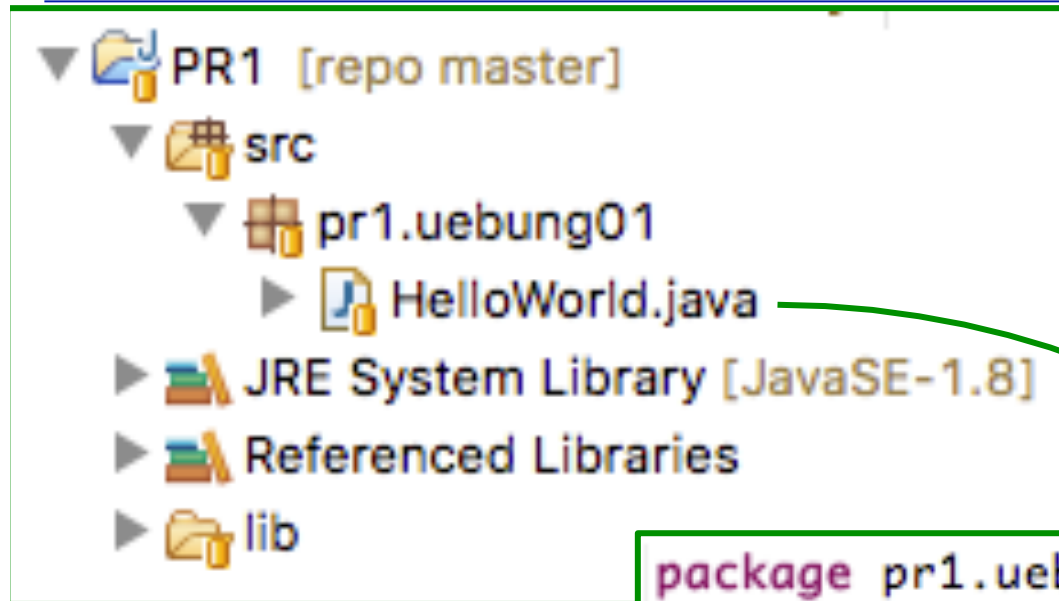
- Sprechen Sie sich mit Ihrem Gruppenpartner ab:
Sie sollten *nicht gleichzeitig an derselben Datei* arbeiten!
- Bevor Sie mit Ihrer Arbeit in Eclipse beginnen, besorgen Sie sich die aktuellste Version aller Dateien des Projekts aus dem zentralen Repository: *Kontextmenü des Projekts -> Team -> Pull*
- Wenn Sie
 - fertig sind,
 - sich an Ihrem Rechner abmelden wollen oder
 - einen stabilen Zwischenstand erreicht haben,

schieben Sie Ihre Änderungen in das zentrale Repository:
Kontextmenü des Projekts -> Team -> Commit: Markieren Sie alle Dateien und geben Sie eine kurze aber aussagekräftige Beschreibung des aktuellen Zwischenstands an, zum Beispiel

- vollständig, aber noch nicht getestet
- noch ein Fehler: negative Zahlen funktionieren nicht
- abgabefertige Version

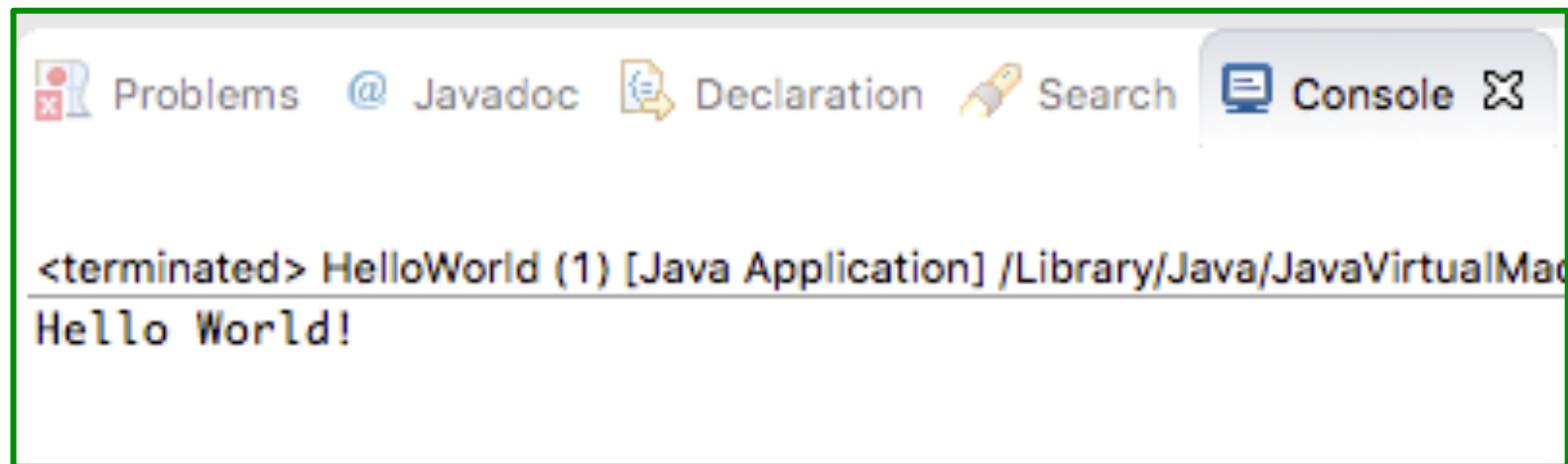
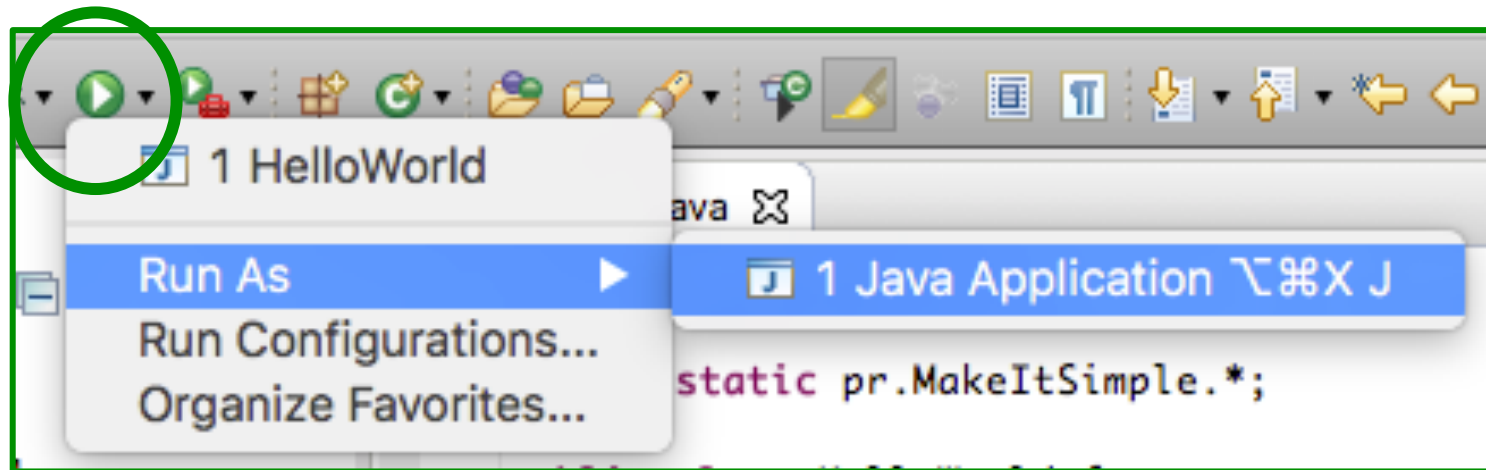
Dann klicken Sie *Commit und Push*, damit Ihre Änderungen ins zentrale Repository übernommen werden

Ein Doppelklick öffnet das Programm im Editor

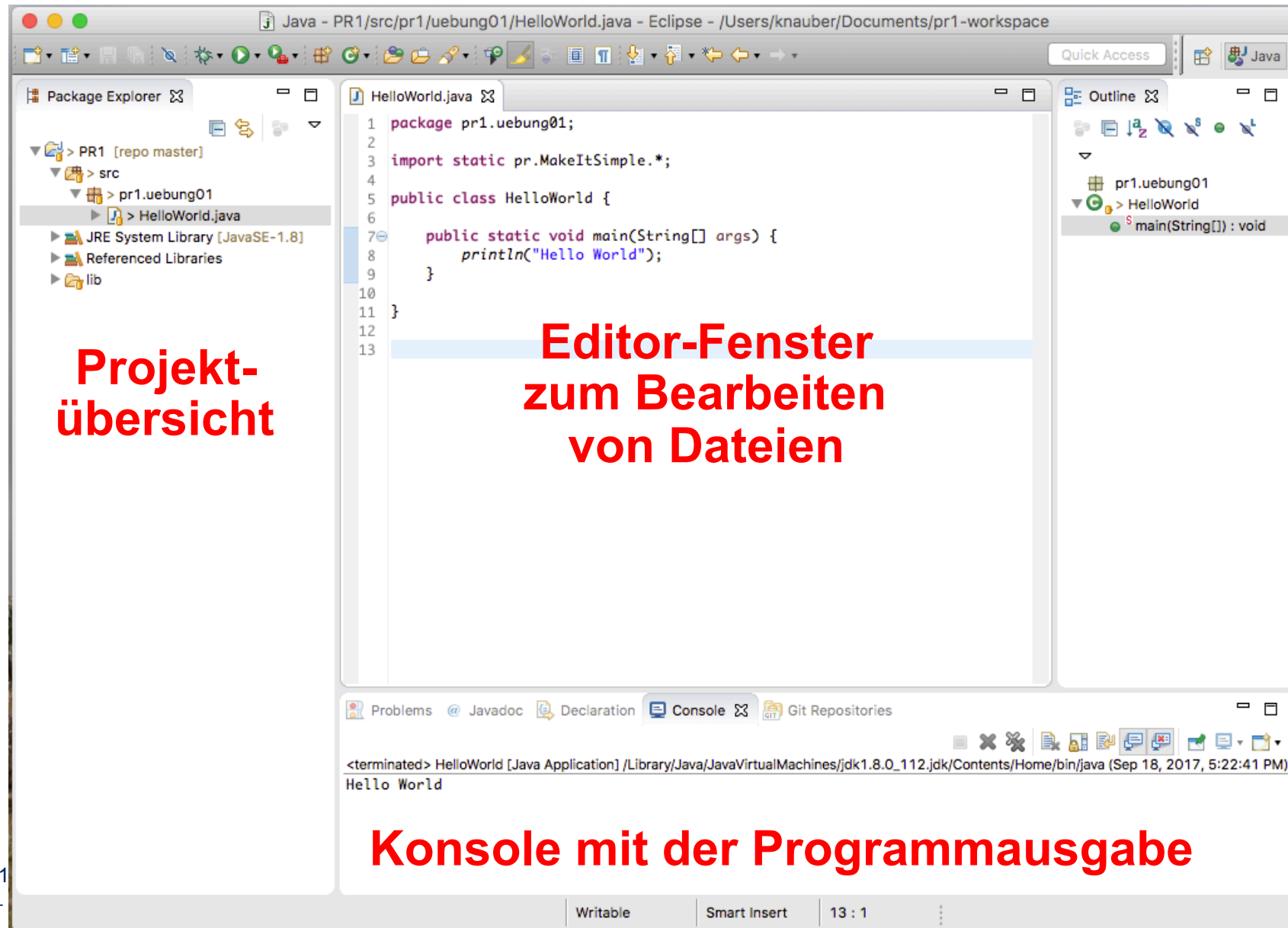


```
package pr1.uebung01;  
  
import static pr.MakeItSimple.*;  
  
public class HelloWorld {  
  
    public static void main(String[] args) {  
        println("Hello World");  
    }  
  
}
```

Start des Programms und Anzeige der Ausgabe in der Konsole



Eclipse in der Übersicht



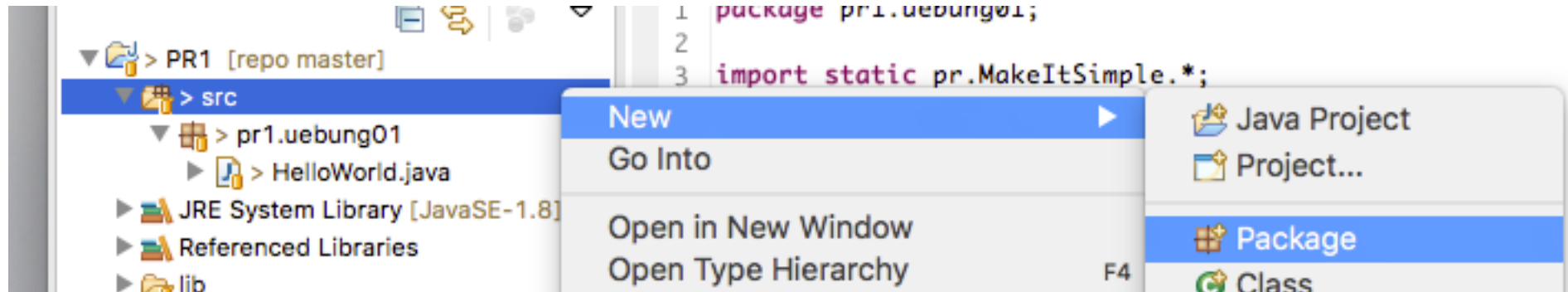
**Projekt-
übersicht**

**Editor-Fenster
zum Bearbeiten
von Dateien**

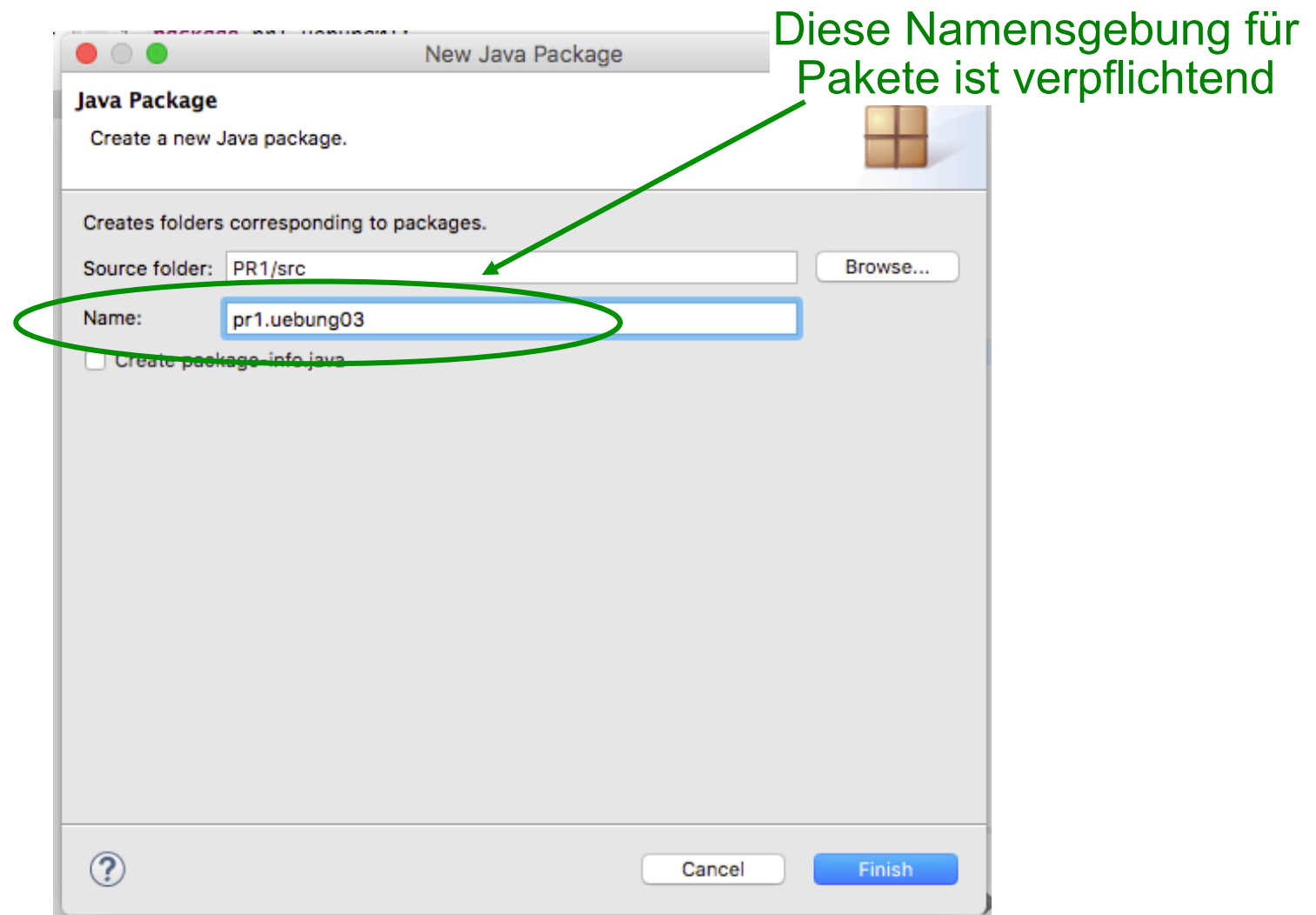
Konsole mit der Programmausgabe

Package anlegen 1/2;

Regel: *1 Package pro Aufgabenblatt*

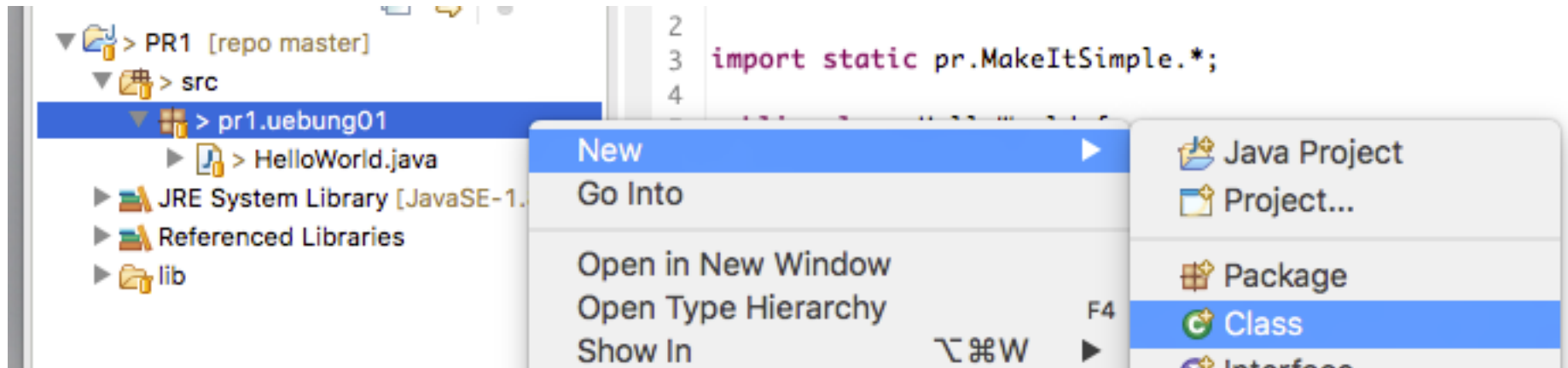


Package anlegen 2/2



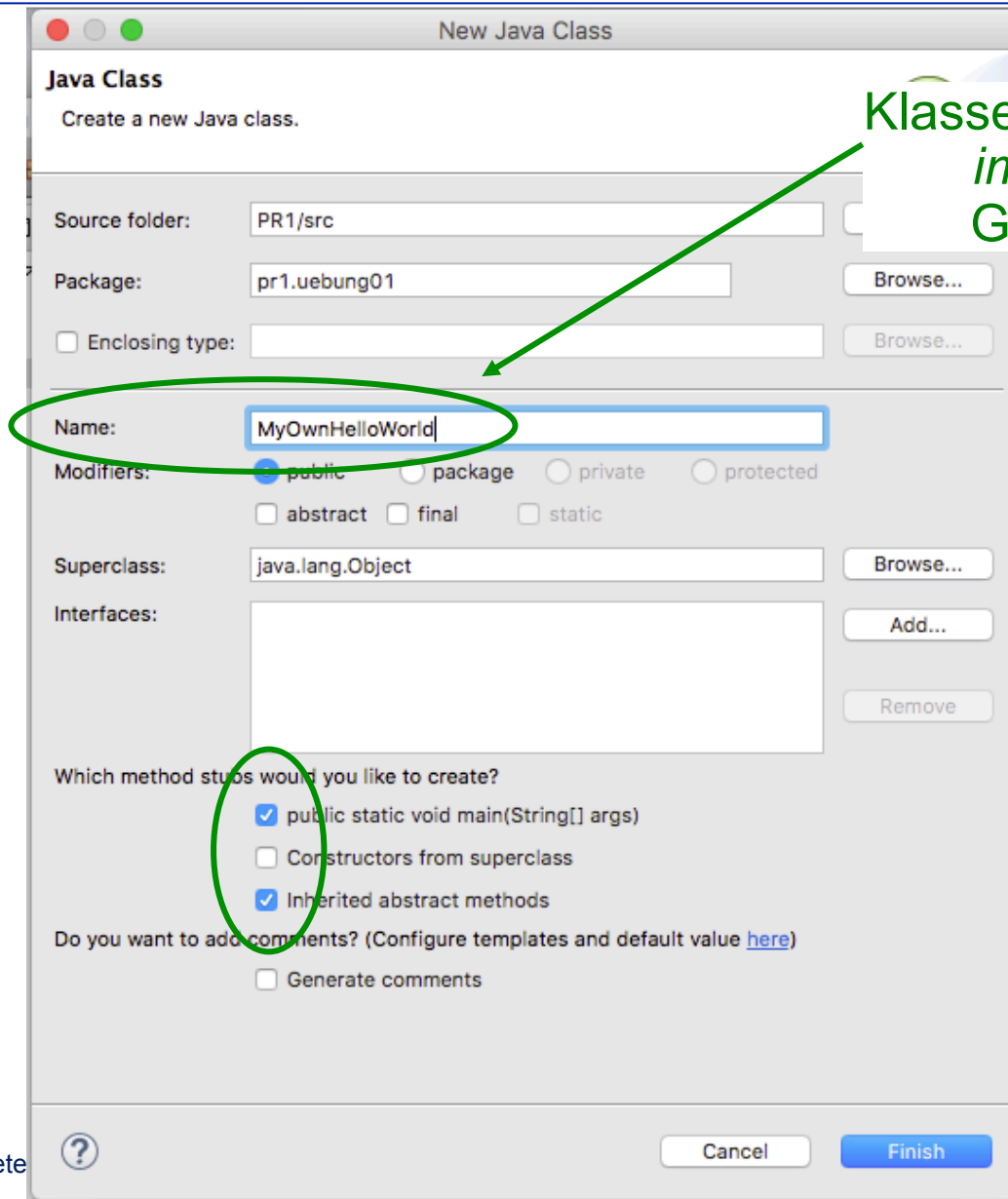
Neues Programm (= Klasse) im Paket anlegen

1/2



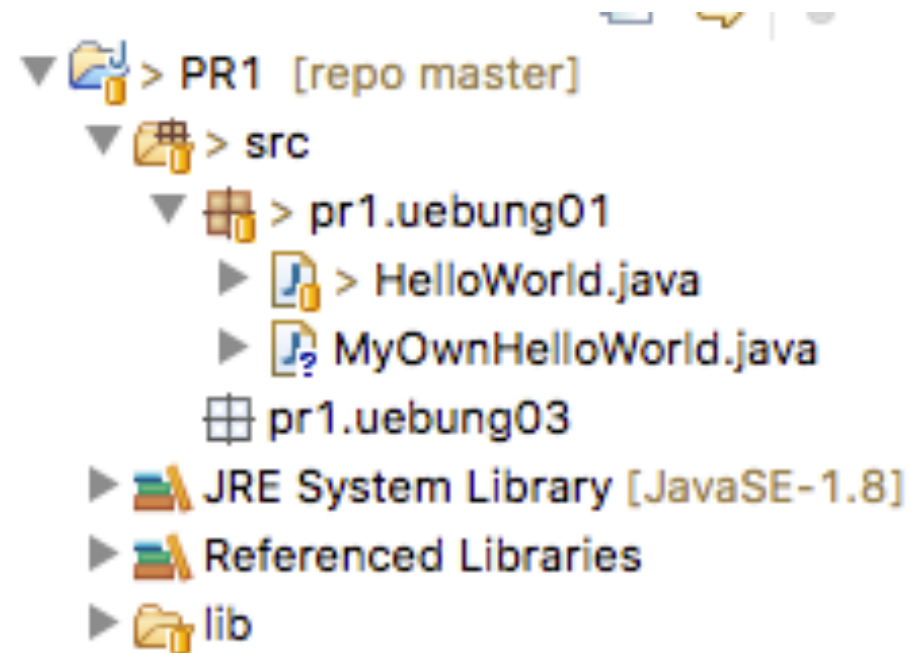
Neues Programm (= Klasse) im Paket anlegen

2/2

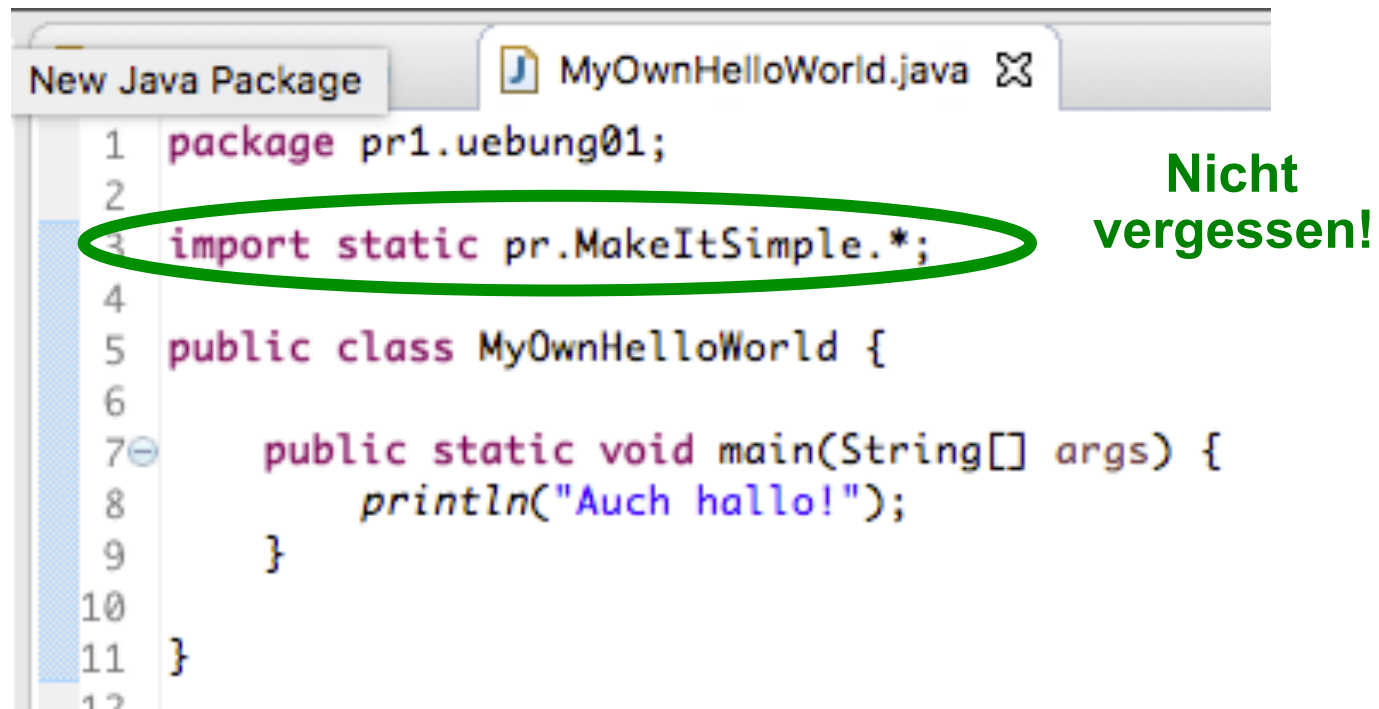


Klassen-Namen beginnen immer mit einem Großbuchstaben

So sieht das Ergebnis aus: unsere neue Klasse und das neue Paket



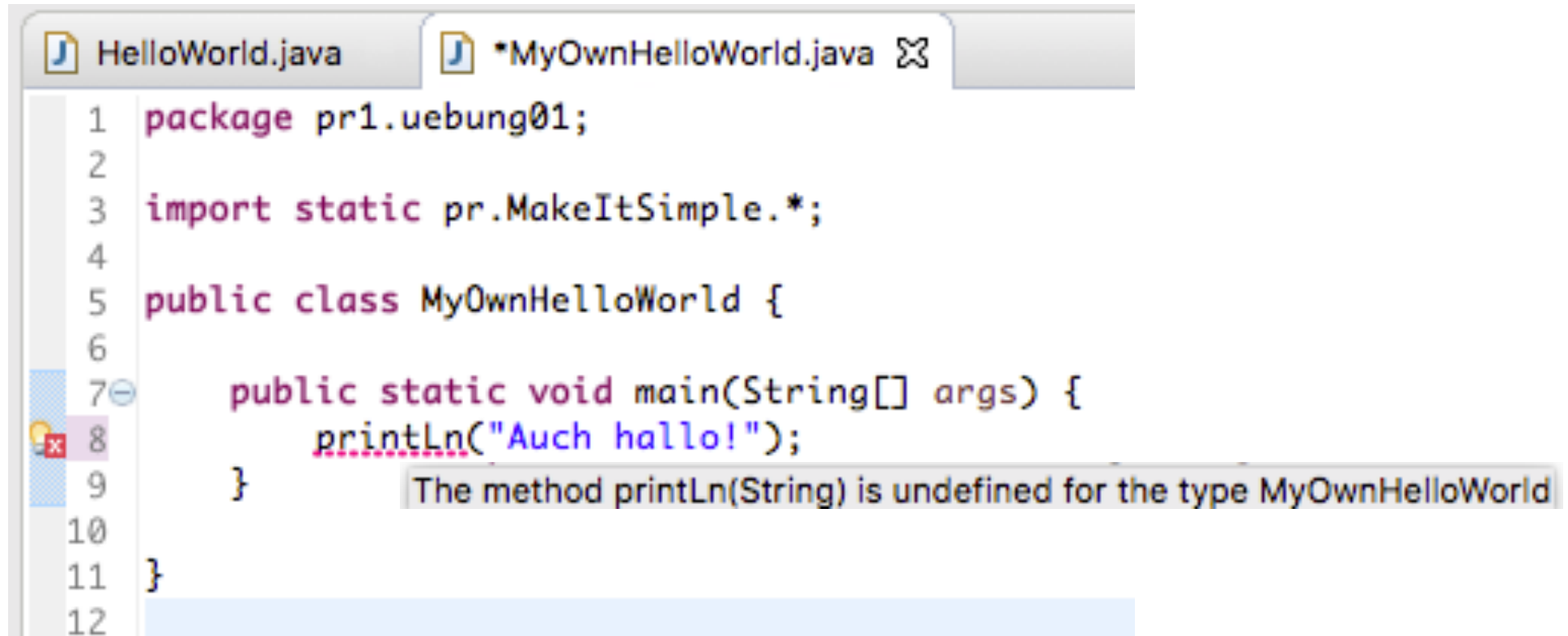
Ein mögliches Programm



```
1 package pr1.uebung01;
2
3 import static pr.MakeItSimple.*;
4
5 public class MyOwnHelloWorld {
6
7     public static void main(String[] args) {
8         println("Auch hallo!");
9     }
10
11 }
12
```

Nicht vergessen!

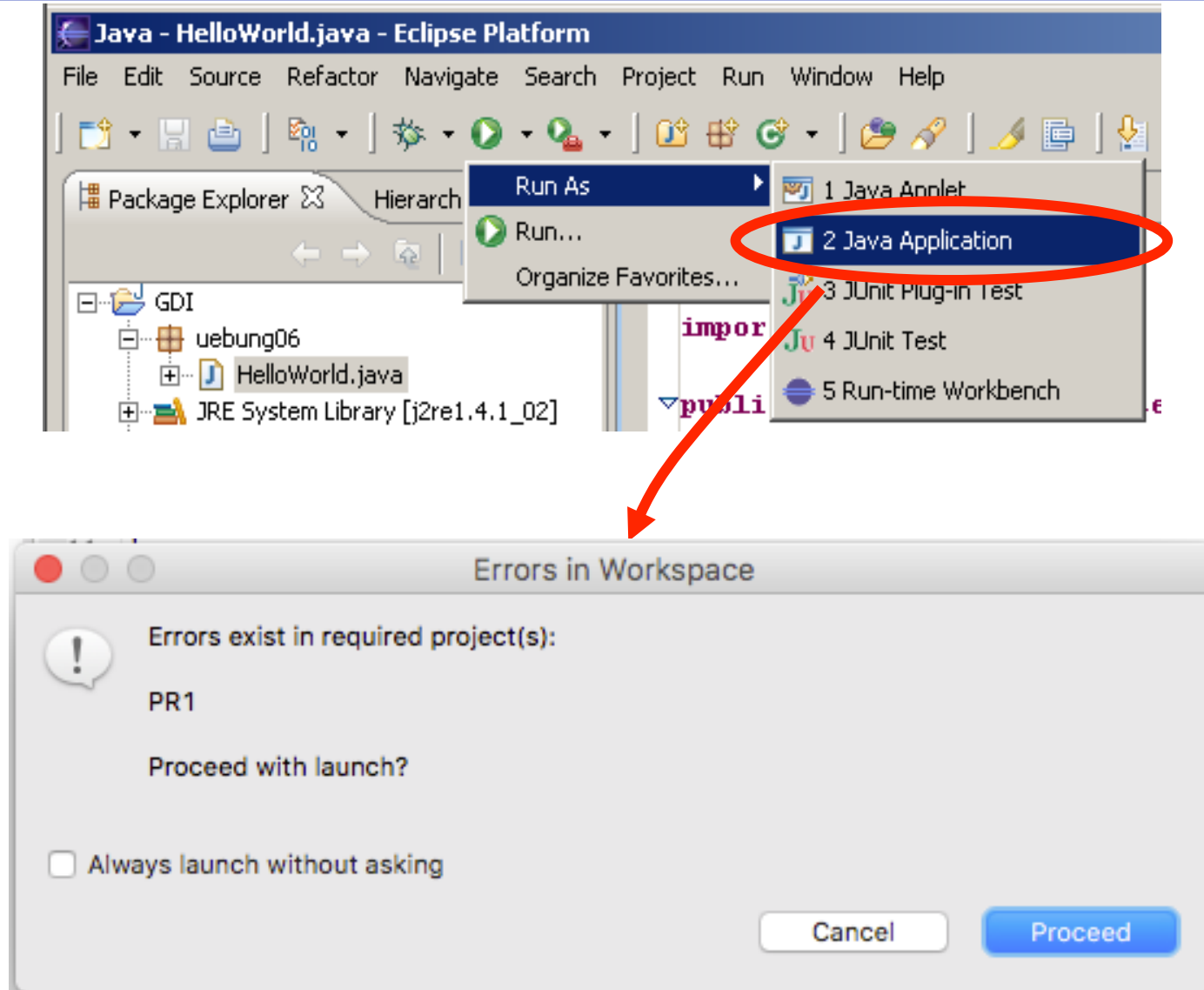
Fehlerhaftes Programm 1/4



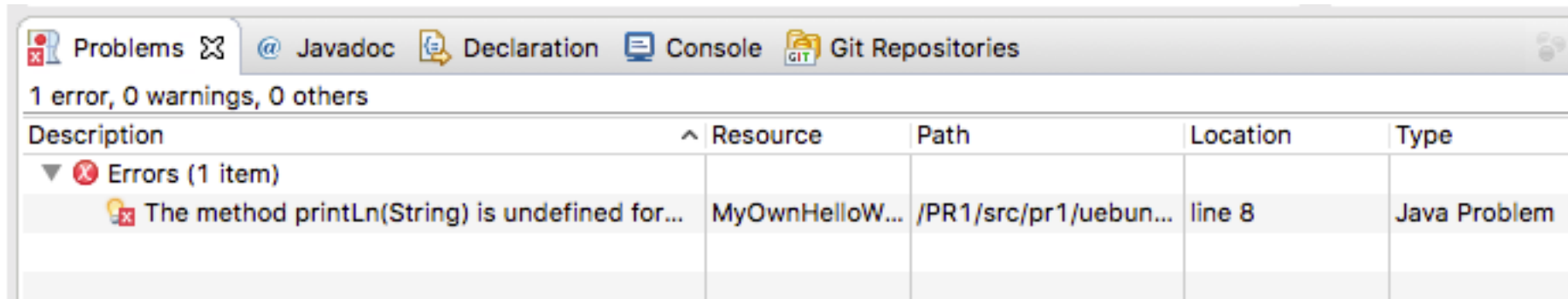
```
1 package pr1.uebung01;
2
3 import static pr.MakeItSimple.*;
4
5 public class MyOwnHelloWorld {
6
7     public static void main(String[] args) {
8         println("Auch hallo!");
9     }
10
11 }
12
```

The method println(String) is undefined for the type MyOwnHelloWorld



Fehlerhaftes Programm 2/4: Übersetzen funktioniert nicht



Fehlerhaftes Programm 3/4: Fehlersuche

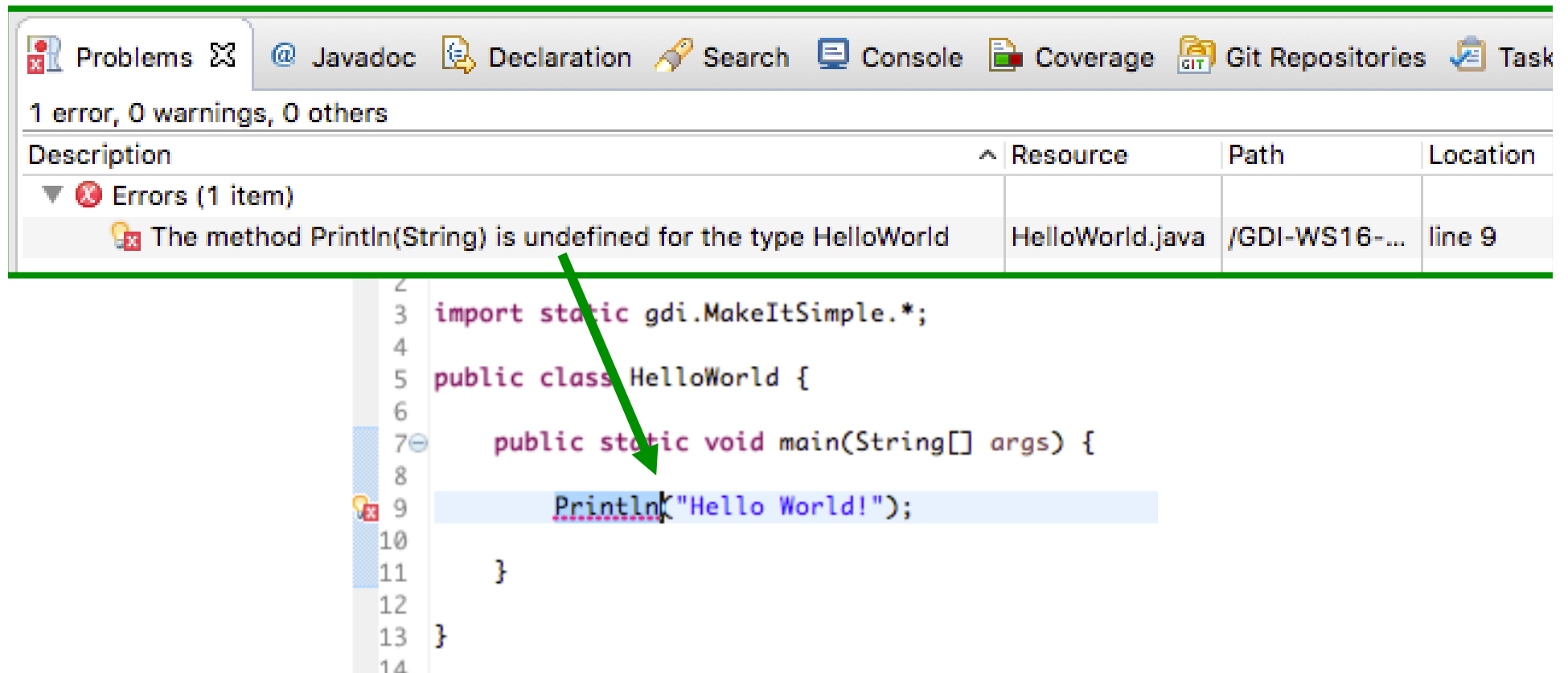


The screenshot shows the Eclipse IDE's Problems view. The title bar includes tabs for Problems, Javadoc, Declaration, Console, and Git Repositories. Below the tabs, it indicates '1 error, 0 warnings, 0 others'. A table lists the error details.

Description	Resource	Path	Location	Type
▼  Errors (1 item)				
 The method println(String) is undefined for...	MyOwnHelloW...	/PR1/src/pr1/uebun...	line 8	Java Problem


Fehlerhaftes Programm 4/4: Fehlersuche

Anklicken der Zeile



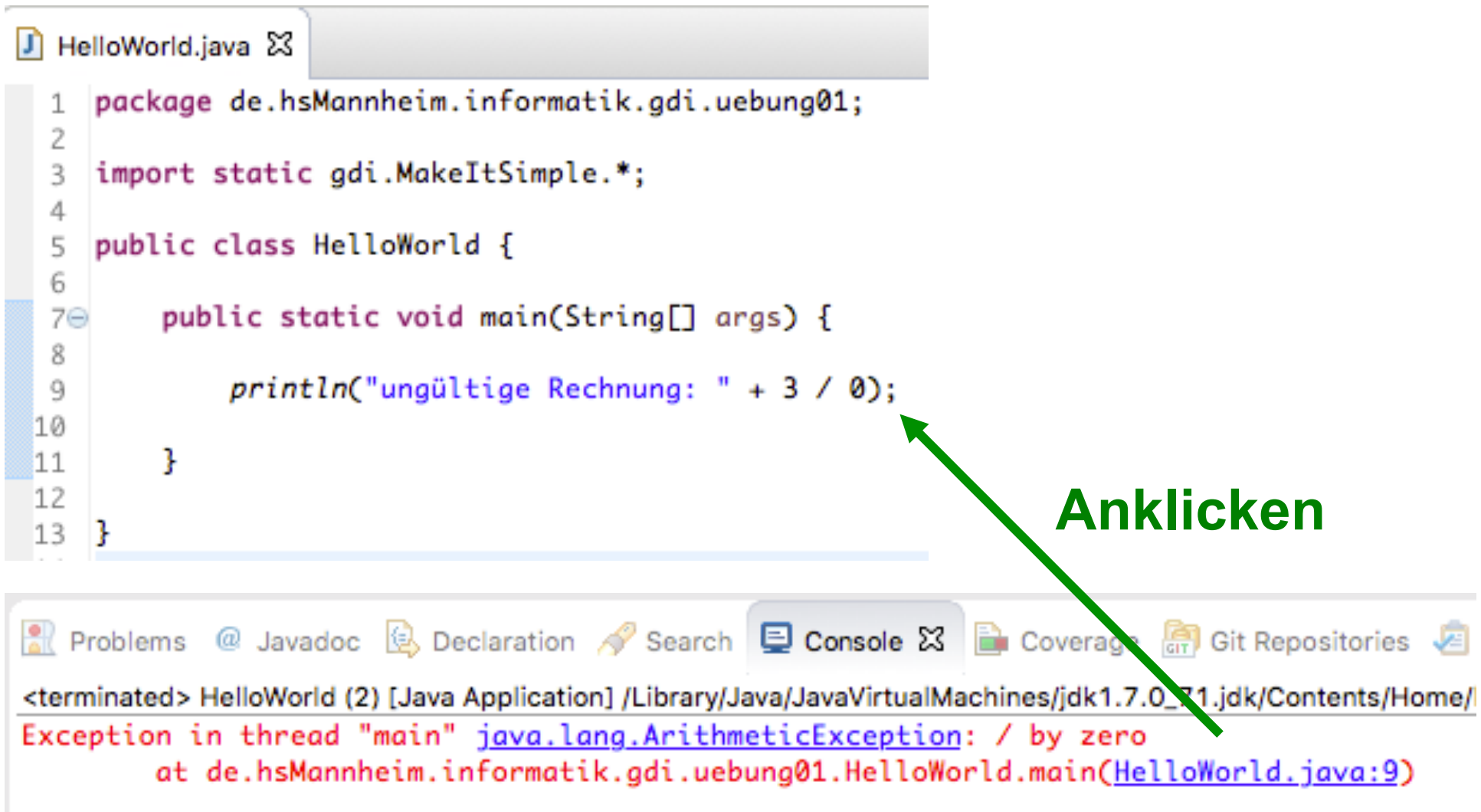
The screenshot shows the Eclipse IDE interface. At the top, the 'Problems' view is active, displaying a table of errors. Below the table, the source code for 'HelloWorld.java' is shown in the editor. A green arrow points from the error message in the Problems view to the line of code in the editor that caused the error.

1 error, 0 warnings, 0 others

Description	Resource	Path	Location
▼  Errors (1 item)			
 The method <code>Println(String)</code> is undefined for the type <code>HelloWorld</code>	HelloWorld.java	/GDI-WS16-...	line 9

```
2  
3 import static gdi.MakeItSimple.*;  
4  
5 public class HelloWorld {  
6  
7     public static void main(String[] args) {  
8  
9         Println("Hello World!");  
10  
11     }  
12  
13 }  
14
```

Laufzeitfehler auf der Konsole



```
1 package de.hsMannheim.informatik.gdi.uebung01;
2
3 import static gdi.MakeItSimple.*;
4
5 public class HelloWorld {
6
7     public static void main(String[] args) {
8
9         println("ungültige Rechnung: " + 3 / 0);
10
11     }
12
13 }
```

Problems @ Javadoc Declaration Search Console Coverage Git Repositories

<terminated> HelloWorld (2) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_71.jdk/Contents/Home/
Exception in thread "main" java.lang.ArithmeticException: / by zero
at de.hsMannheim.informatik.gdi.uebung01.HelloWorld.main(HelloWorld.java:9)

Anklicken

Haben Sie Fragen?