

# Migrating to eclipse

RoboCode Special Interest Topic

Summer 2011



**Christian Riess, Eva Eibenberger**

**Pattern Recognition Lab (Computer Science Dep. 5)**

**Friedrich-Alexander-University Erlangen-Nuremberg**

# Eclipse – eine integrierte Entwicklungsumgebung



- Wir haben gesehen, dass es egal ist, wie wir den Code schreiben – Hauptsache, er steht am Ende in einer Datei
- Integrierte Entwicklungsumgebungen (IDE, „integrated development environment“):
  - Assistieren bei dem Schreiben des Codes
  - Assistieren bei der Fehlersuche
- Eclipse ist eine IDE, die auf Java-Code spezialisiert ist
  - Syntax-Prüfung während der Eingabe (z.B. vergessene Strichpunkte)
  - Liste verfügbarer Methoden für ein Objekt während der Eingabe
  - Roboter kann aus der Umgebung heraus kompiliert und ausgeführt werden

# Umzug des Roboters in ein „Eclipse Projekt“

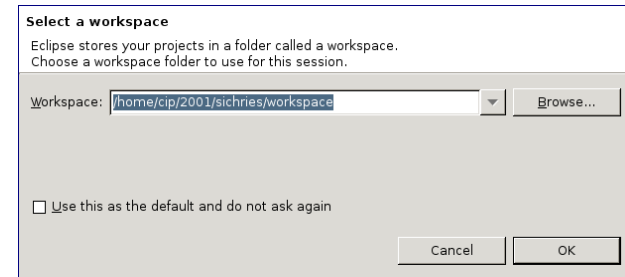


- Klassen sind in Eclipse in „Projekten“ organisiert
- Die RoboCode-Bibliotheken müssen diesem Projekt bekannt gemacht werden
- Wie RoboCode ausgeführt wird, muss dem Projekt bekannt gemacht werden
  
- (steht auch alles auf  
[http://robowiki.net/wiki/Robocode/Eclipse/Create\\_a\\_Project](http://robowiki.net/wiki/Robocode/Eclipse/Create_a_Project)  
[http://robowiki.net/w/index.php?title=Robocode/Running\\_from\\_Eclipse](http://robowiki.net/w/index.php?title=Robocode/Running_from_Eclipse)  
)
  
- Im Cip:  
Eclipse ausführen: `eclipse`

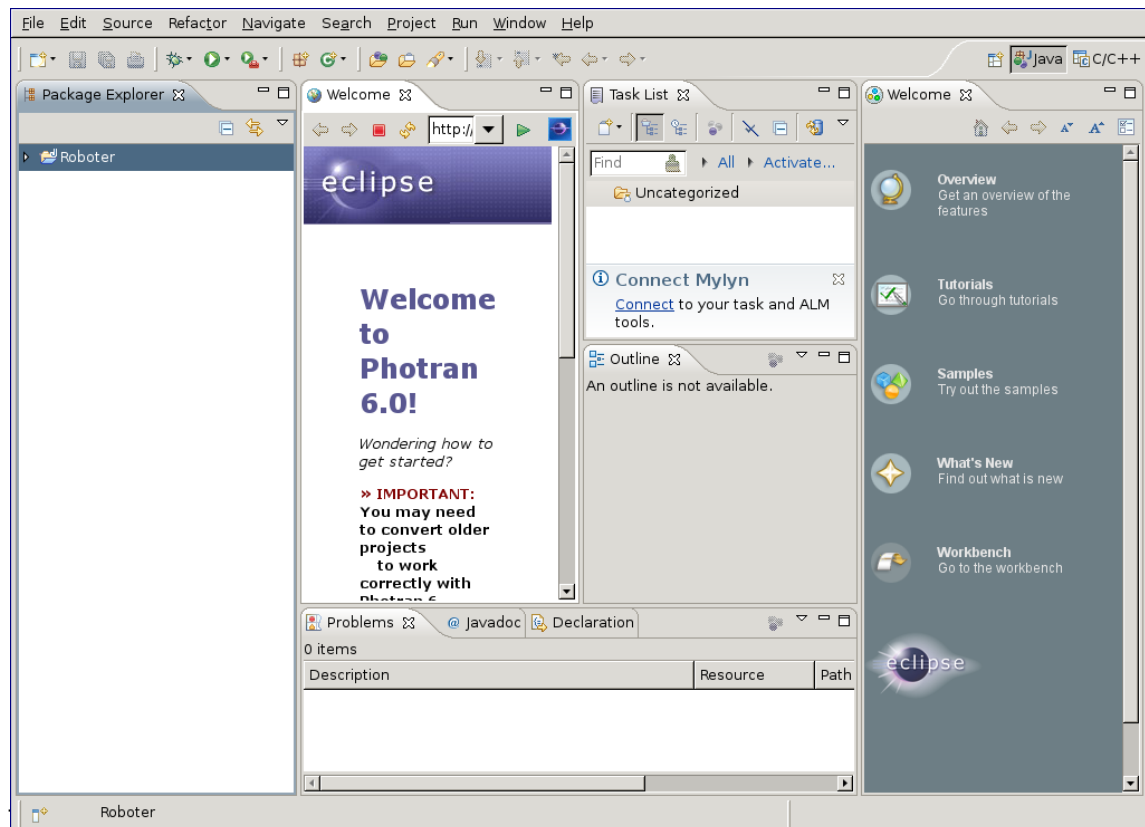
# Step by step



- Der vorgeschlagene Workspace sollte OK sein



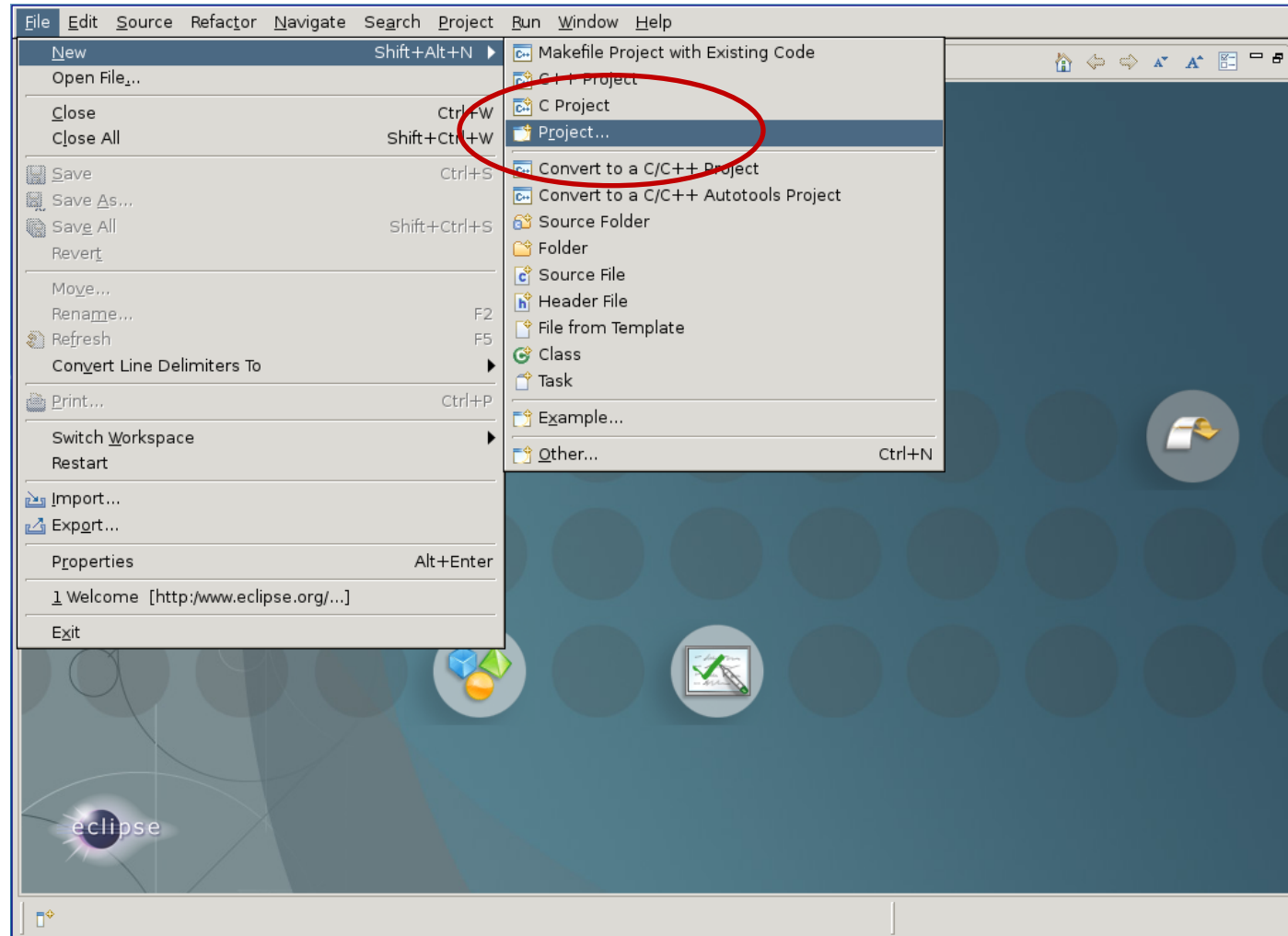
- Eventuell muss man das Fenster vergrößern, so wie abgebildet ist es zu überladen.



# Step by step



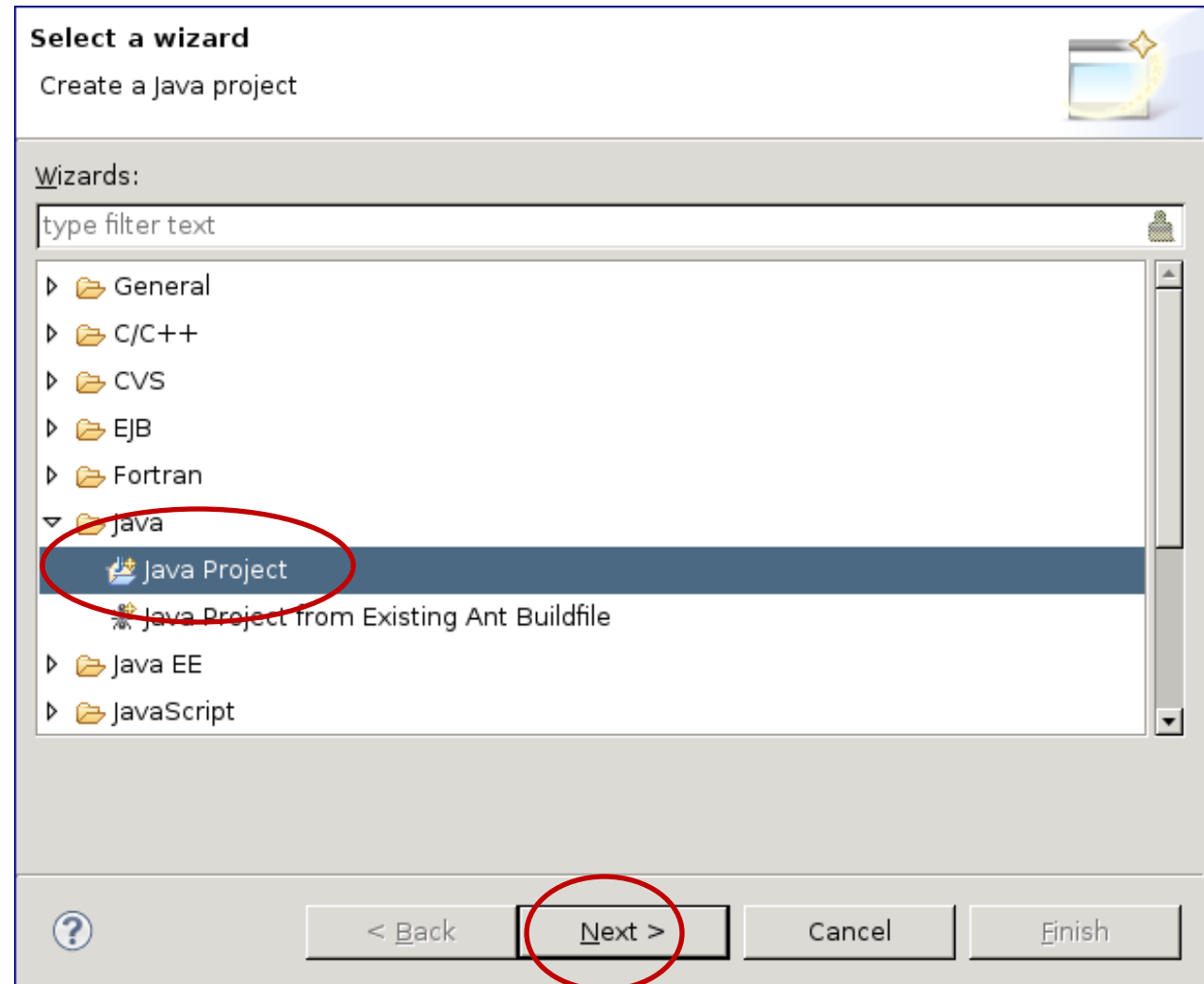
- Code ist in Eclipse in „Projekten“ organisiert, also erst ein neues Projekt anlegen:



# Step by step



- Der Projekttyp ist „Java Project“





# Step by step

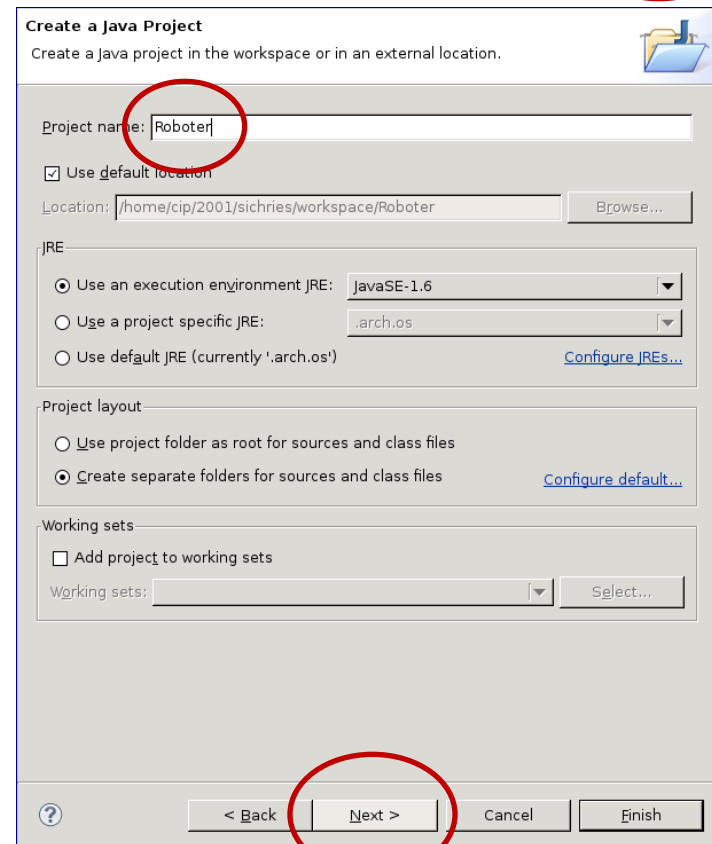
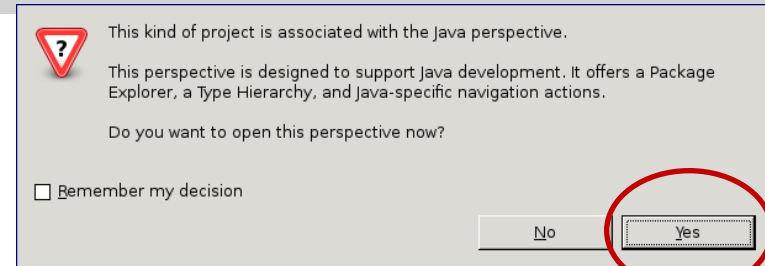
- Noch eine Frage positiv beantworten (ja, öffne die Java-Ansicht)

- Und einen Projektnamen eingeben.

Der Name ist eigentlich egal, aber:

- keine Umlaute
- keine Leerstellen
- ...dann klappt alles

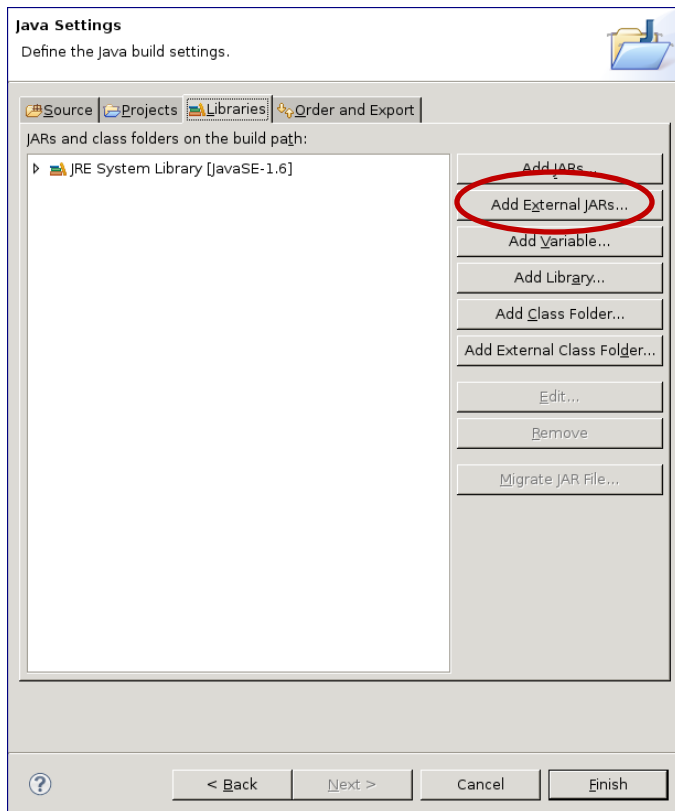
- -> "Next" klicken



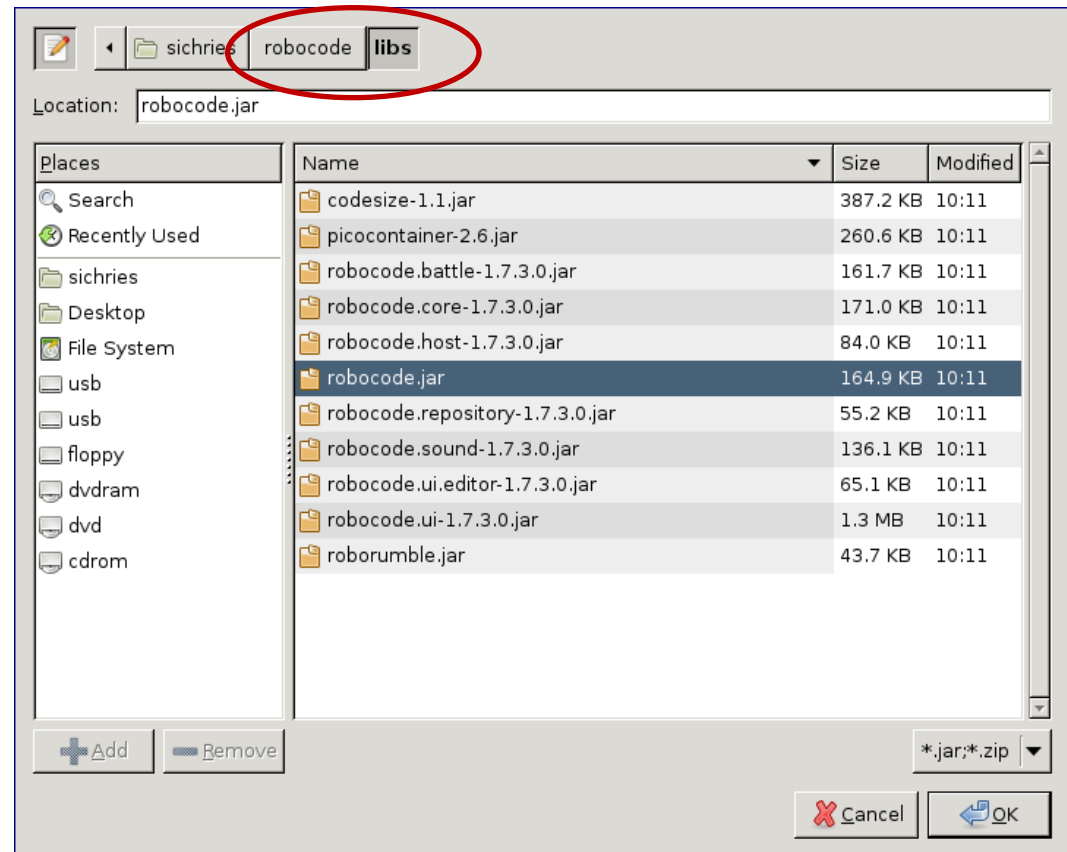


# Step by step

- Wir müssen der Umgebung noch die RoboCode-Bibliothek bekannt machen



- Konkret muss das Java Archiv robocode.jar (aus robocode/libs/) hinzugefügt werden.

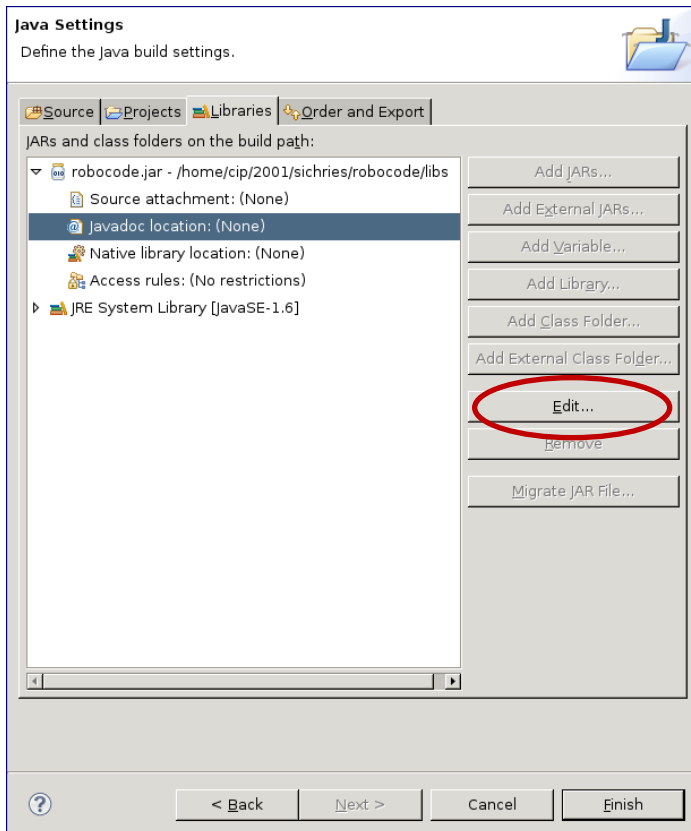




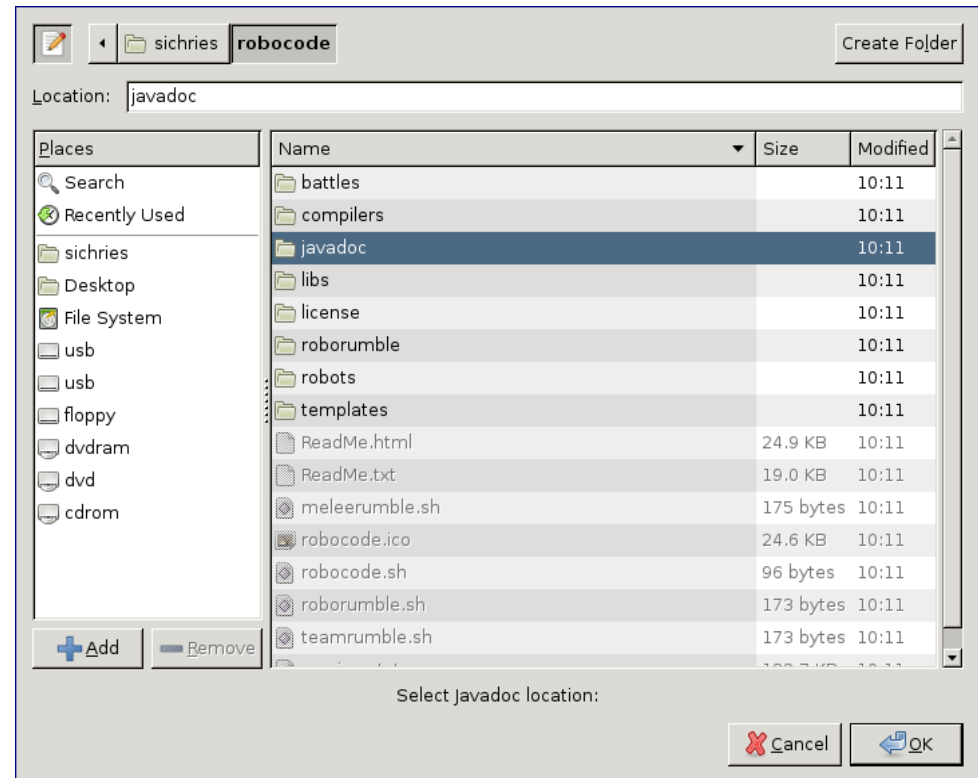


# Step by step

- Zu der Library kann man noch die Dokumentation registrieren:



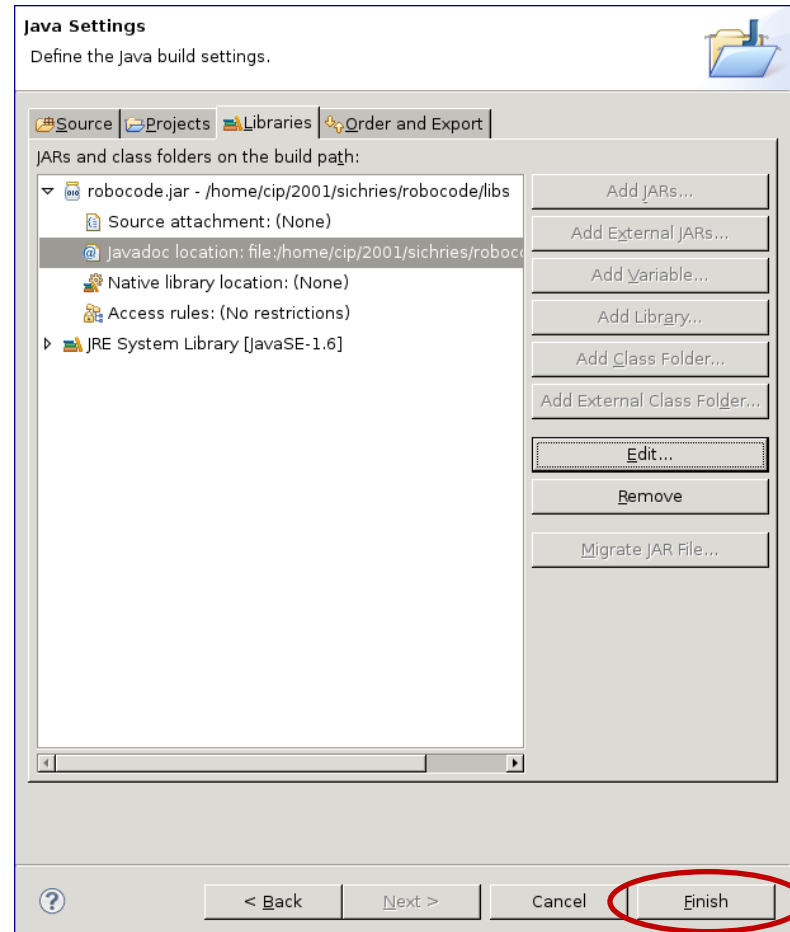
- Dafür das Unterverzeichnis robocode/javadoc angeben



# Step by step



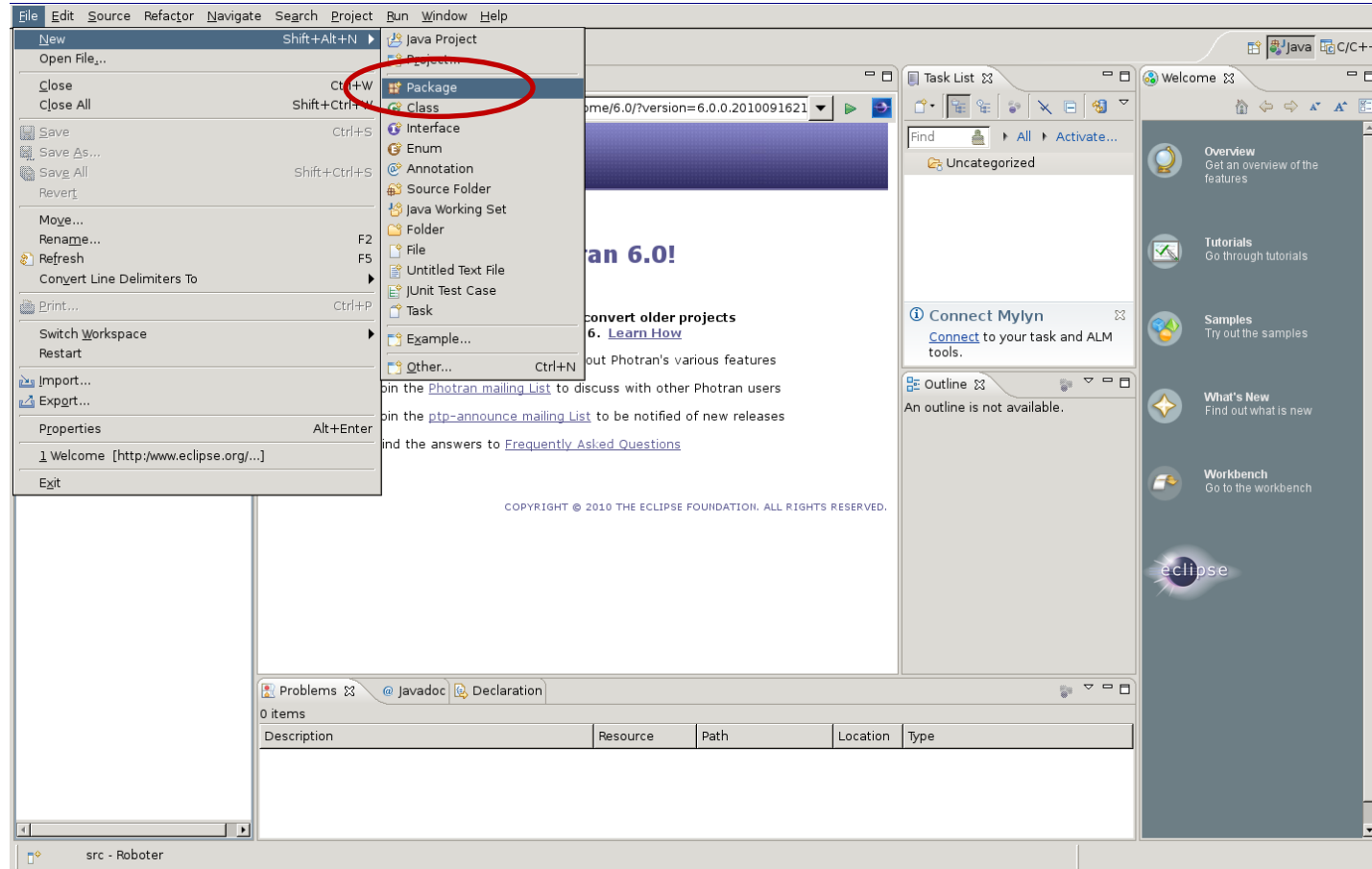
- ...dann ist das Projekt konfiguriert!





# Step by step

- Dann ein neues package anlegen (ähnlich wie in dem RoboCode-Editor)





# Step by step

- Und in dem package eine neue Klasse...

The screenshot shows the Eclipse IDE interface. The 'New' menu is open, and the 'Class' option is highlighted with a red circle. In the Project Explorer on the left, the package 'chriess' is also highlighted with a red circle. The main editor area displays a document titled 'Welcome [http://www.eclipse.org/...]' with the text 'Eclipse 6.0!' and 'convert older projects'.

At the bottom of the IDE, there are two panels showing 'Problems' and 'Tasks'.

**Problems Panel:**

Description	Resource	Path	Location	Type
0 items				

**Tasks Panel:**

Description	Resource	Path	Location	Type
0 items				

# Step by step

- Den Roboter-Namen...
- ...und die Oberklasse (robocode.Robot oder robocode.AdvancedRobot) angeben
- Auf „finish“ klicken.

**Java Class**  
Create a new Java class.

Source folder:

Package:

Enclosing type:

---

Name:

Modifiers:  public  default  private  protected  
 abstract  final  static

Superclass:

Interfaces:

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

# Step by step



- ...und coden!
- ...zum Schluss muss man noch einstellen, wie das Programm ausgeführt wird (siehe nächste Folie)

```

package chriess;

import robocode.Robot;

public class RockHard extends Robot {
}

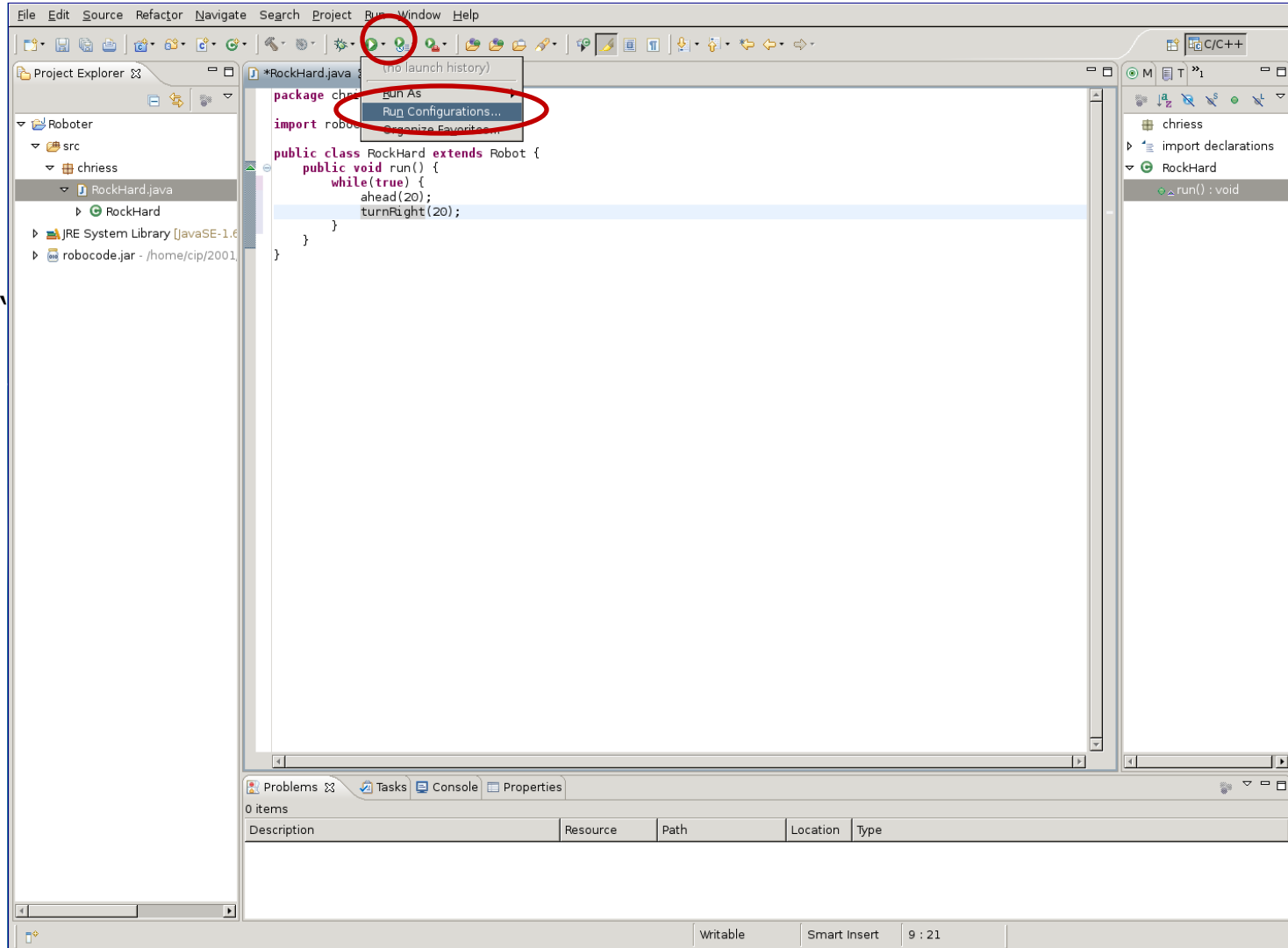
```

The screenshot shows the Eclipse IDE interface. The Project Explorer on the left shows the project structure: Roboter > src > chriess > RockHard.java. The main editor displays the code for RockHard.java. The Package Explorer on the right shows the package chriess and the class RockHard. The bottom status bar indicates 'Writable', 'Smart Insert', and '6 : 1'.



# Step by step

- Konfiguration der Laufzeitumgebung: „Run Configurations“



# Step by step

- Doppelklick auf „Java Application“ erzeugt eine neue Konfiguration

Create, manage, and run configurations

✖ Main type not specified

Name: New\_configuration

Project: Roboter

Main class:

Include system libraries when searching for a main class  
 Include inherited mains when searching for a main class  
 Stop in main

Apply Revert

Close Run



# Step by step

- Die Hauptklasse ist die Turnierarena, robocode.Robocode
- ...danach den Reiter „Arguments“ auswählen...

Create, manage, and run configurations  
Run a Java application

Name: New\_configuration

Project: Roboter

Main class: robocode.Robocode

Include system libraries when searching for a main class  
 Include inherited mains when searching for a main class  
 Stop in main

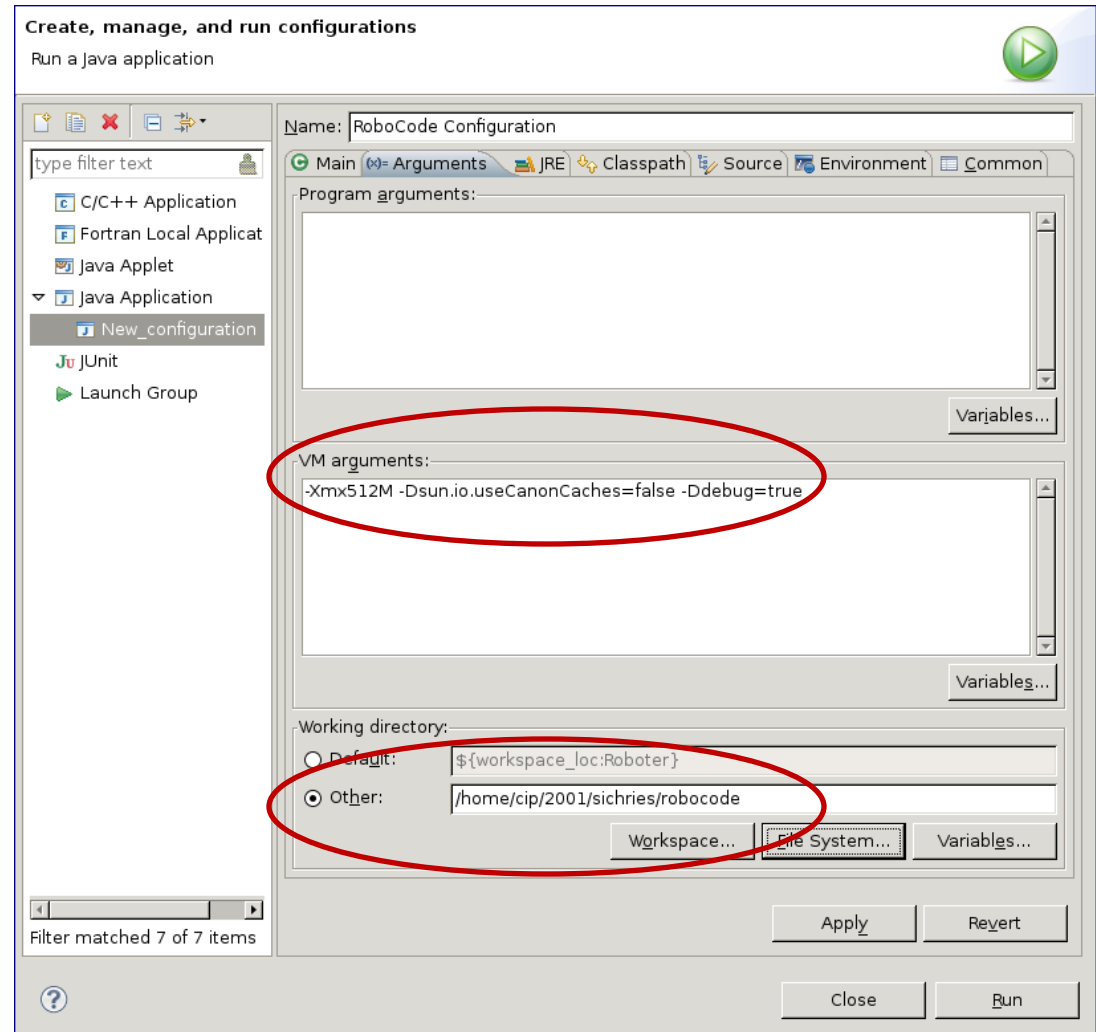
Apply Revert

Close Run



# Step by step

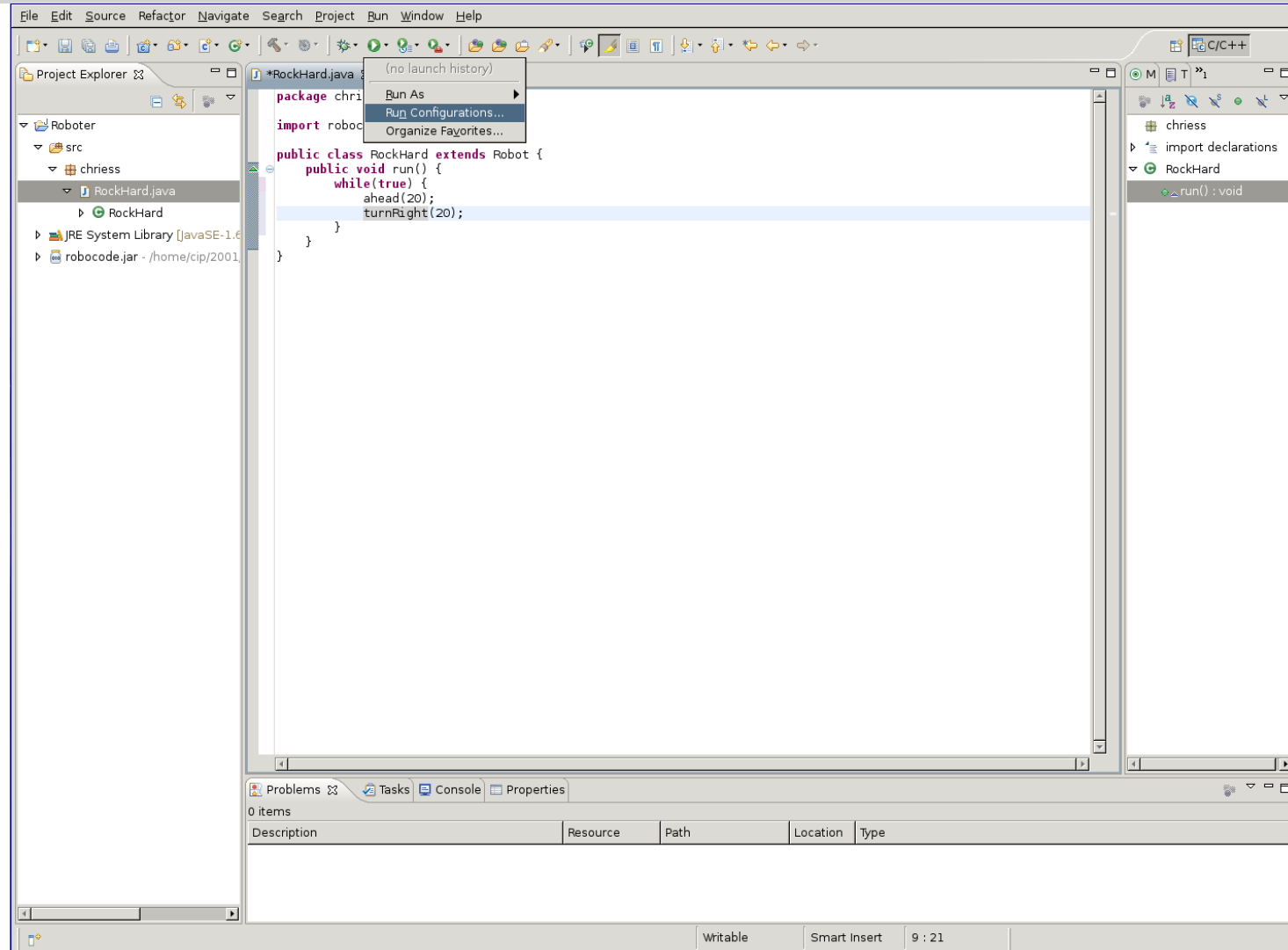
- ...und die Argumente abtippen.
- Wichtig ist außerdem, das Arbeitsverzeichnis auf das RoboCode-Verzeichnis zu setzen



# Step by step



- Fertig!
- Zum Ausführen auf den grünen Knopf drücken!



# Step by step

- In der RoboCode-Arena muss abschließend noch der Speicherort der eclipse-Roboter bekannt gemacht werden
- Die Klassen sind normalerweise in `~/workspace/<Projektname>/bin`

