Migrating to eclipse

RoboCode Special Interest Topic

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 compiliert und ausgeführt werden

Umzug des Roboters in ein „Eclipse Projekt"

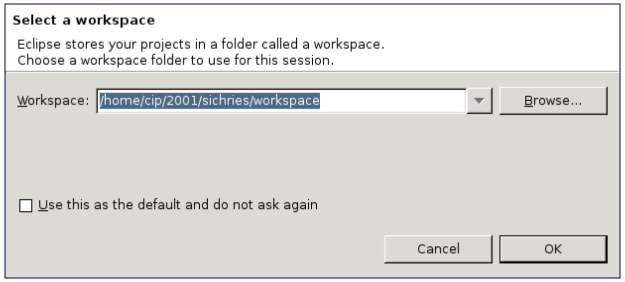
- Klassen sind in Eclipse in „Projekten" organisiert
- Die RoboCode-Bibliotheken müssen diesem Projekt bekannt gemacht werden
- Wie RoboCode ausgeführt wird, muss dem Projekt bekannt gemacht werden

- (steht auch alles auf
  http://robowiki.net/wiki/Robocode/Eclipse/Create_a_Project
  http://robowiki.net/w/index.php?title=Robocode/Running_from_Eclipse
)

- Im Cip:
  Eclipse ausführen: **`eclipse`**

Step by step

- Der vorgeschlagene Workspace sollte OK sein



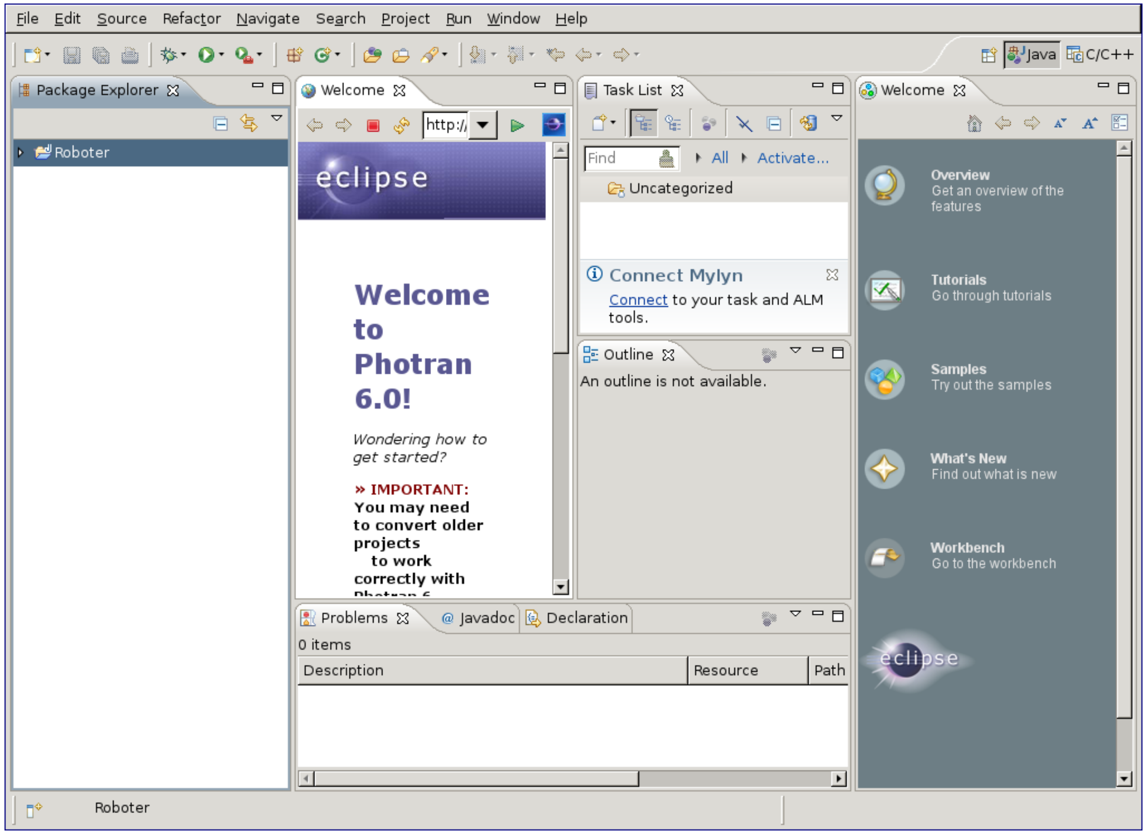
Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: /home/cip/2001/sichries/workspace

☐ Use this as the default and do not ask again

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



Welcome to Photran 6.0!

Wondering how to get started?

» IMPORTANT: You may need to convert older projects to work correctly with Photran 6

Connect to your task and ALM tools.

An outline is not available.

0 items

Description	Resource	Path

Overview – Get an overview of the features

Tutorials – Go through tutorials

Samples – Try out the samples

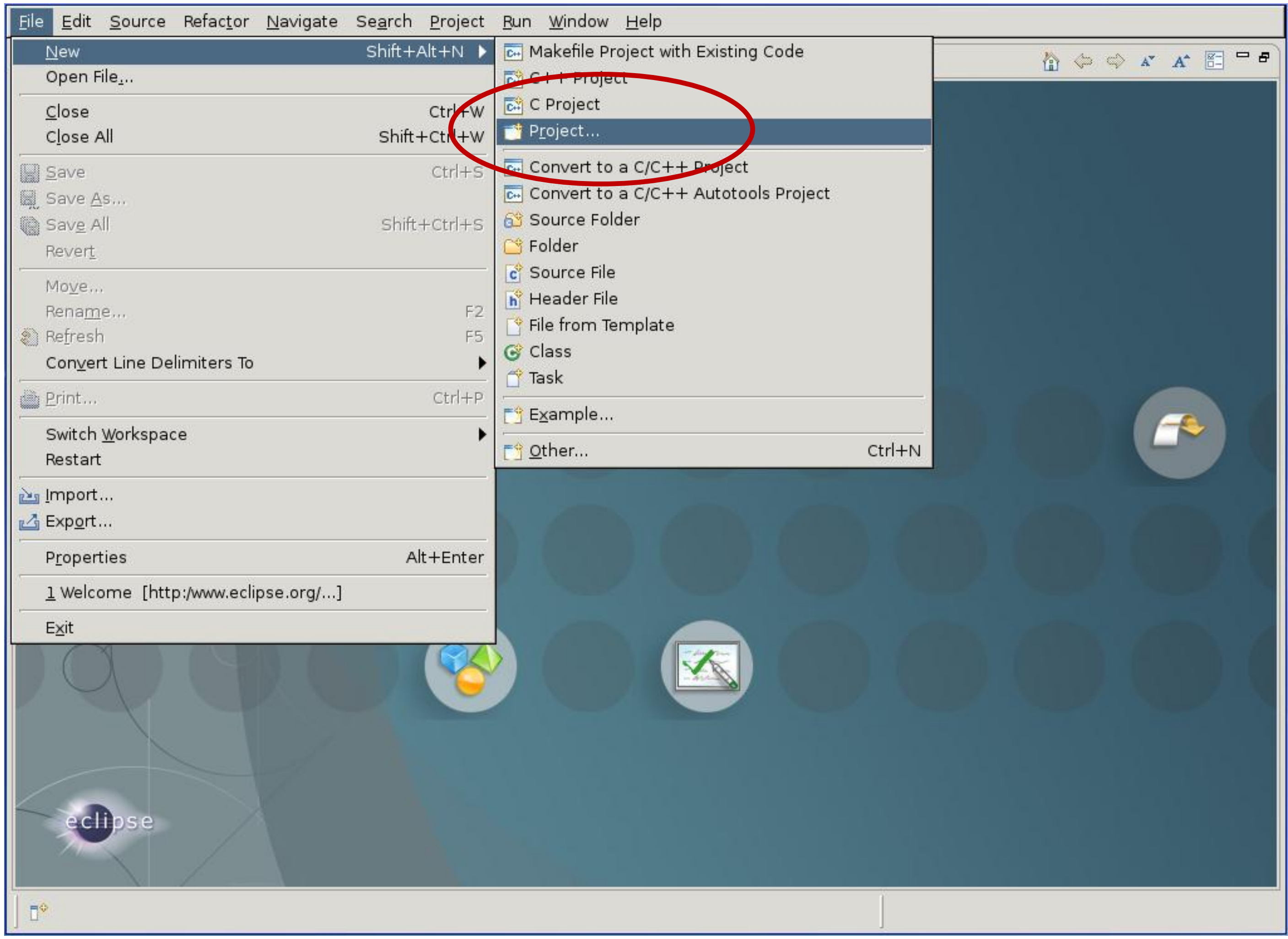
What's New – Find out what is new

Workbench – Go to the workbench

Roboter

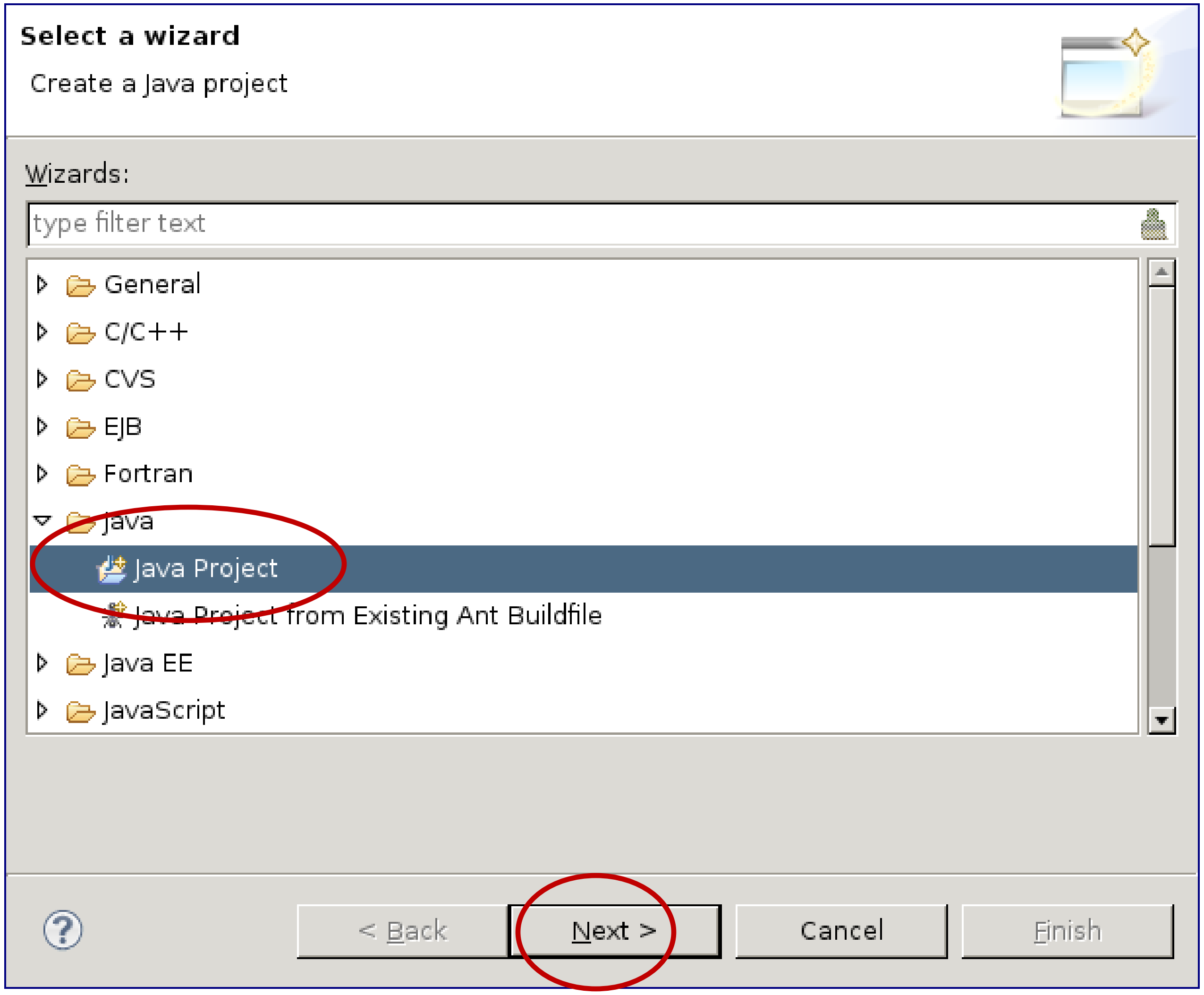
Step by step

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



Step by step

- Der Projekttyp ist „Java Project“



Select a wizard

Create a Java project

Wizards:

type filter text

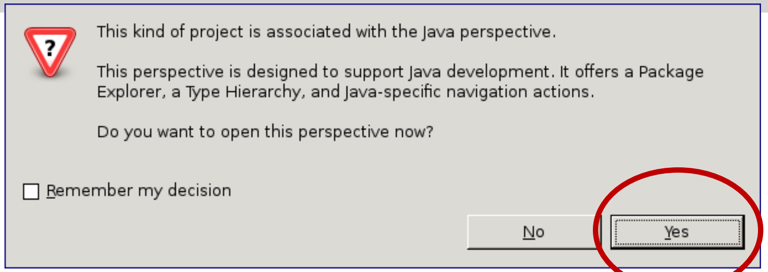
- General
- C/C++
- CVS
- EJB
- Fortran
- Java
  - Java Project
  - Java Project from Existing Ant Buildfile
- Java EE
- JavaScript

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" clicken

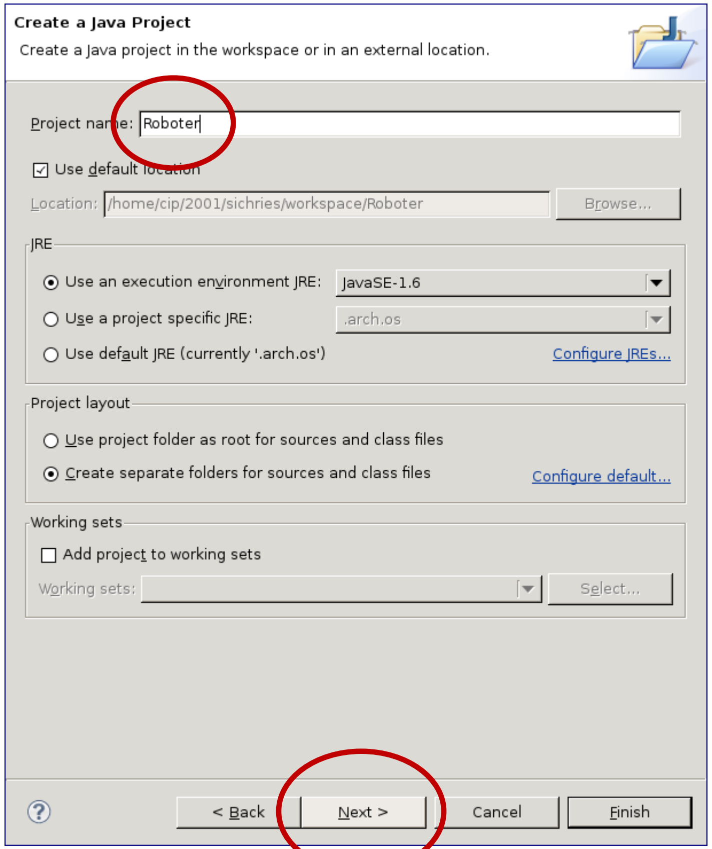


This kind of project is associated with the Java perspective.

This perspective is designed to support Java development. It offers a Package Explorer, a Type Hierarchy, and Java-specific navigation actions.

Do you want to open this perspective now?

☐ Remember my decision



Create a Java Project

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

Project name: Roboter

☑ Use default location

Location: /home/cip/2001/sichries/workspace/Roboter

JRE

◉ Use an execution environment JRE: JavaSE-1.6

○ Use a project specific JRE: .arch.os

○ Use default JRE (currently '.arch.os')

Project layout

○ Use project folder as root for sources and class files

◉ Create separate folders for sources and class files

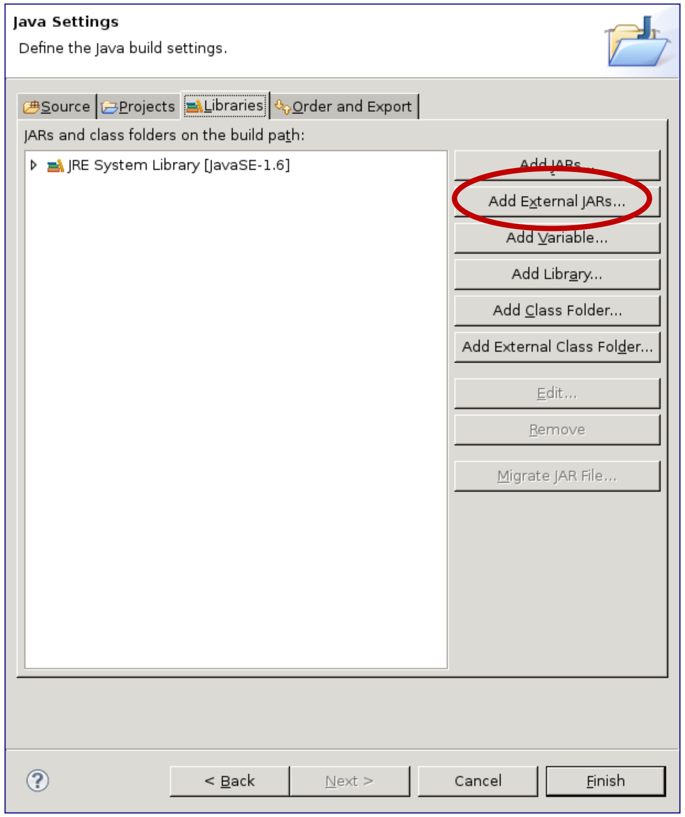
Working sets

☐ Add project to working sets

Working sets:

Step by step

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



Java Settings

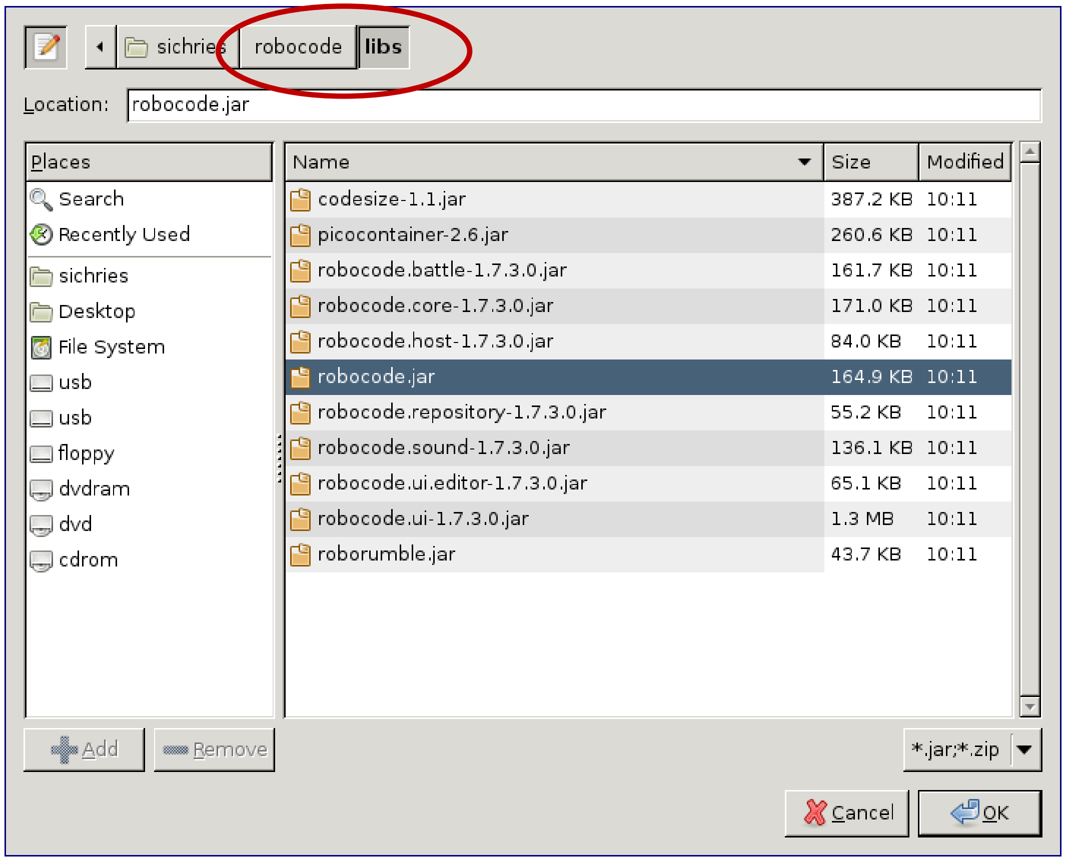
Define the Java build settings.

Source | Projects | Libraries | Order and Export

JARs and class folders on the build path:

JRE System Library [JavaSE-1.6]

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



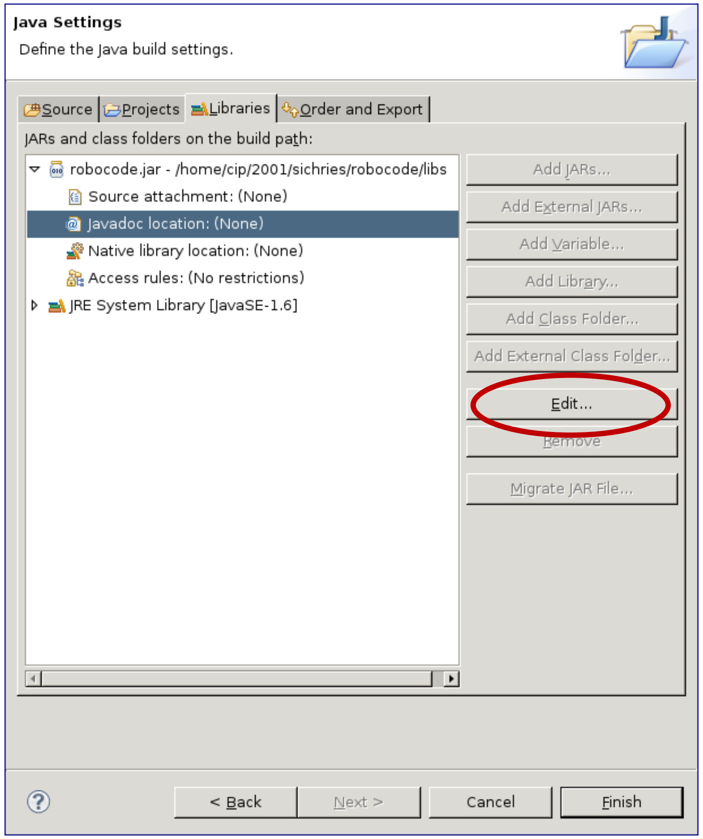
sichries | robocode | libs

Location: robocode.jar

Name	Size	Modified
codesize-1.1.jar	387.2 KB	10:11
picocontainer-2.6.jar	260.6 KB	10:11
robocode.battle-1.7.3.0.jar	161.7 KB	10:11
robocode.core-1.7.3.0.jar	171.0 KB	10:11
robocode.host-1.7.3.0.jar	84.0 KB	10:11
robocode.jar	164.9 KB	10:11
robocode.repository-1.7.3.0.jar	55.2 KB	10:11
robocode.sound-1.7.3.0.jar	136.1 KB	10:11
robocode.ui.editor-1.7.3.0.jar	65.1 KB	10:11
robocode.ui-1.7.3.0.jar	1.3 MB	10:11
roborumble.jar	43.7 KB	10:11

Step by step

- Zu der Library kann man noch die Dokumentation registrieren:



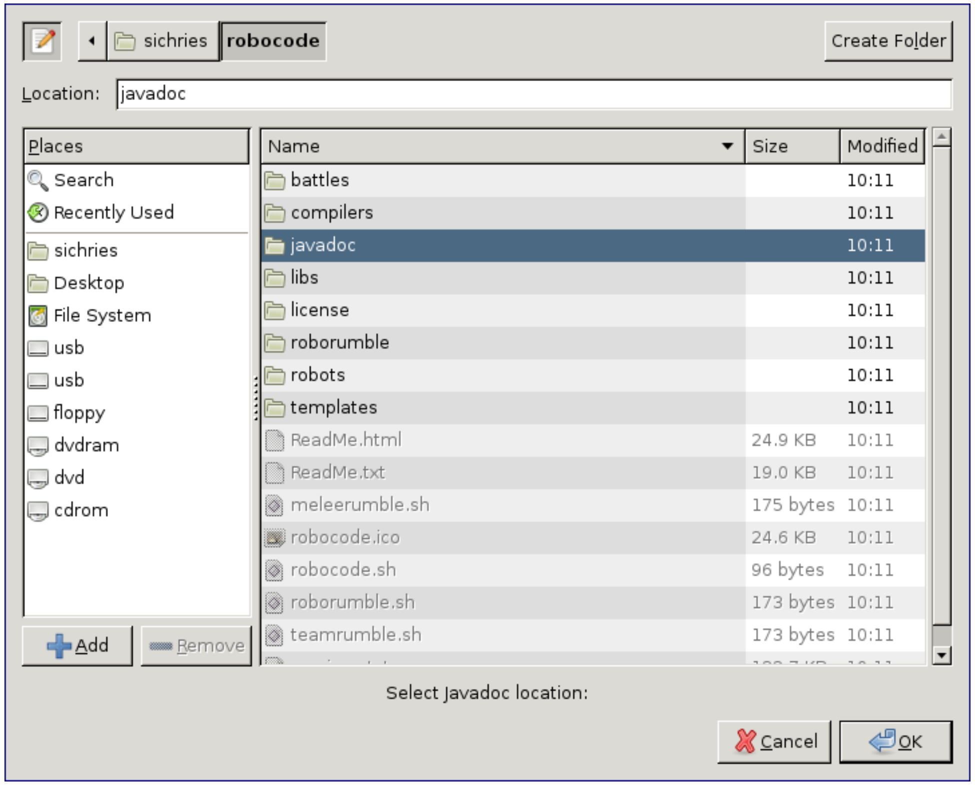
Java Settings
Define the Java build settings.

Source | Projects | Libraries | Order and Export

JARs and class folders on the build path:

- robocode.jar - /home/cip/2001/sichries/robocode/libs
  - Source attachment: (None)
  - Javadoc location: (None)
  - Native library location: (None)
  - Access rules: (No restrictions)
- JRE System Library [JavaSE-1.6]

- Dafür das Unterverzeichnis robocode/javadoc angeben



sichries | robocode

Location: javadoc

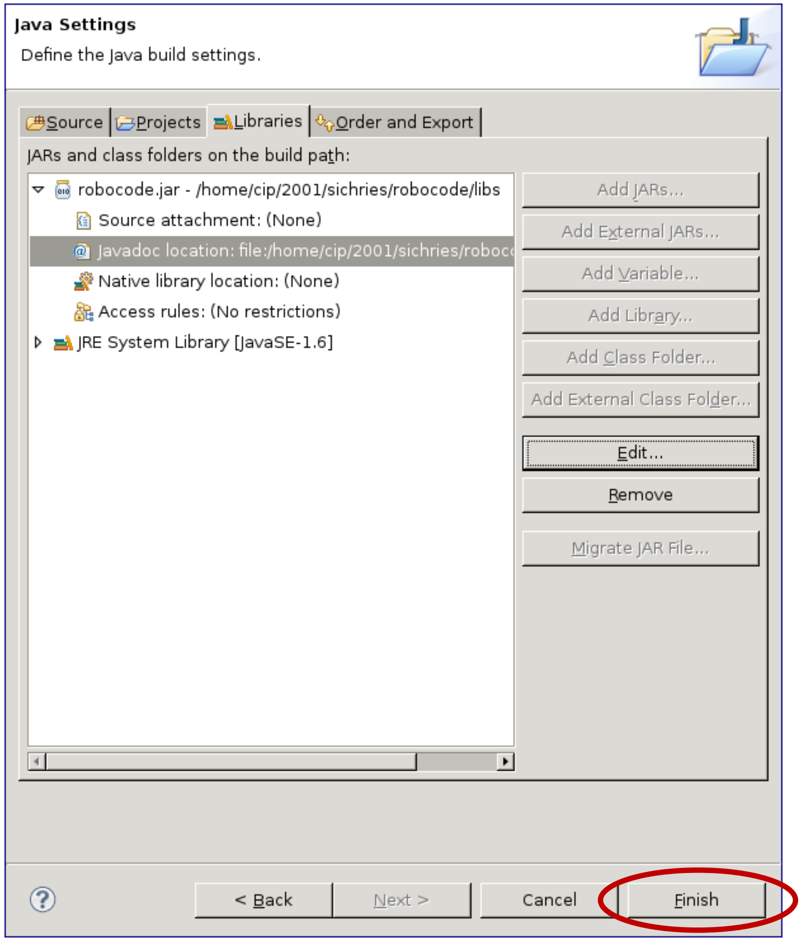
Places
Search
Recently Used
sichries
Desktop
File System
usb
usb
floppy
dvdram
dvd
cdrom

Name	Size	Modified
battles		10:11
compilers		10:11
javadoc		10:11
libs		10:11
license		10:11
roborumble		10:11
robots		10:11
templates		10:11
ReadMe.html	24.9 KB	10:11
ReadMe.txt	19.0 KB	10:11
meleerumble.sh	175 bytes	10:11
robocode.ico	24.6 KB	10:11
robocode.sh	96 bytes	10:11
roborumble.sh	173 bytes	10:11
teamrumble.sh	173 bytes	10:11

Select Javadoc location:

Step by step

- …dann ist das Projekt konfiguriert!



Java Settings
Define the Java build settings.

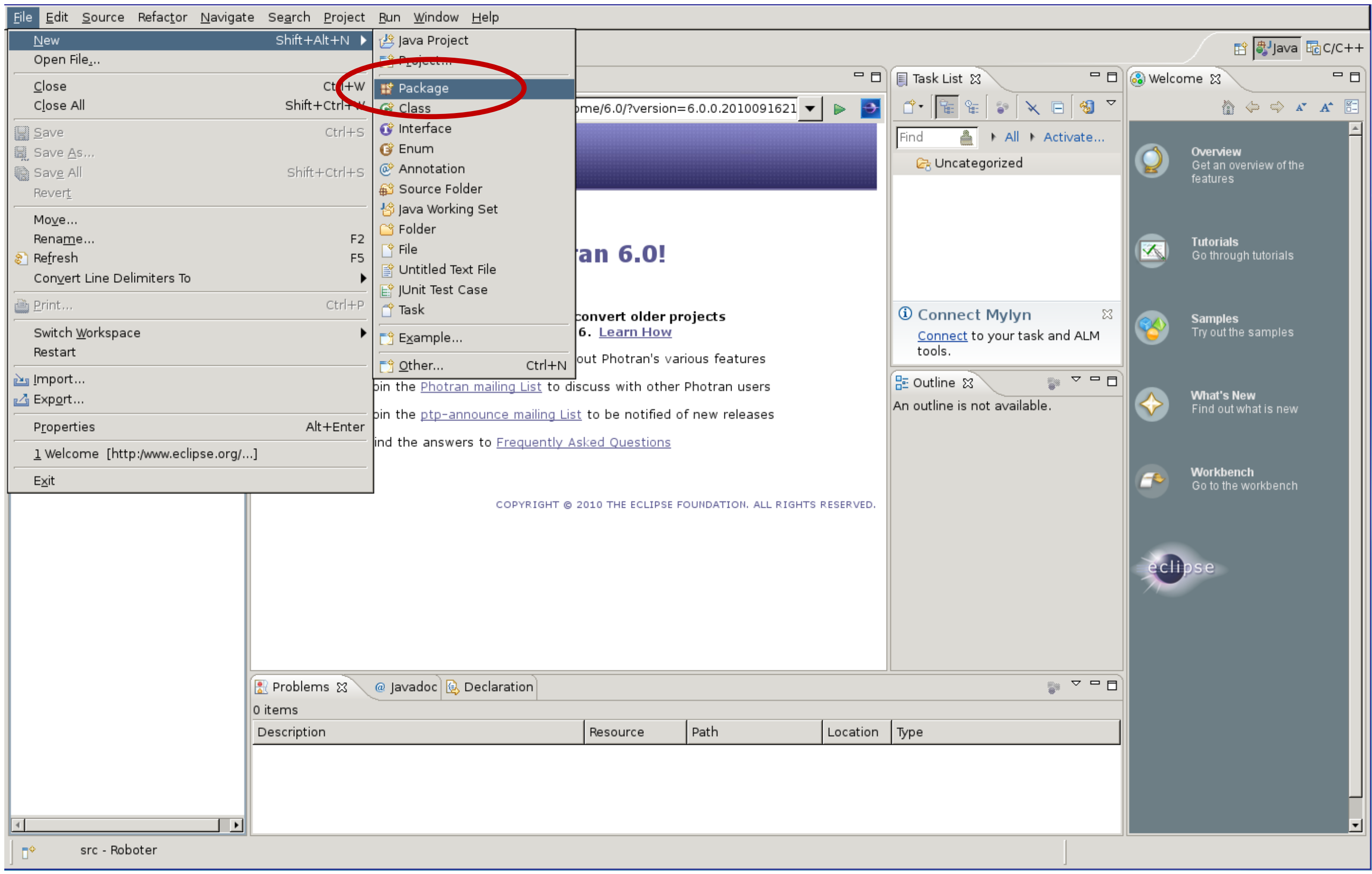
Source | Projects | Libraries | Order and Export

JARs and class folders on the build path:

- robocode.jar - /home/cip/2001/sichries/robocode/libs
  - Source attachment: (None)
  - Javadoc location: file:/home/cip/2001/sichries/roboc
  - Native library location: (None)
  - Access rules: (No restrictions)
- JRE System Library [JavaSE-1.6]

Step by step

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



ome/6.0/?version=6.0.0.2010091621

ran 6.0!

convert older projects
6. Learn How

out Photran's various features

oin the Photran mailing List to discuss with other Photran users

oin the ptp-announce mailing List to be notified of new releases

ind the answers to Frequently Asked Questions

COPYRIGHT © 2010 THE ECLIPSE FOUNDATION. ALL RIGHTS RESERVED.

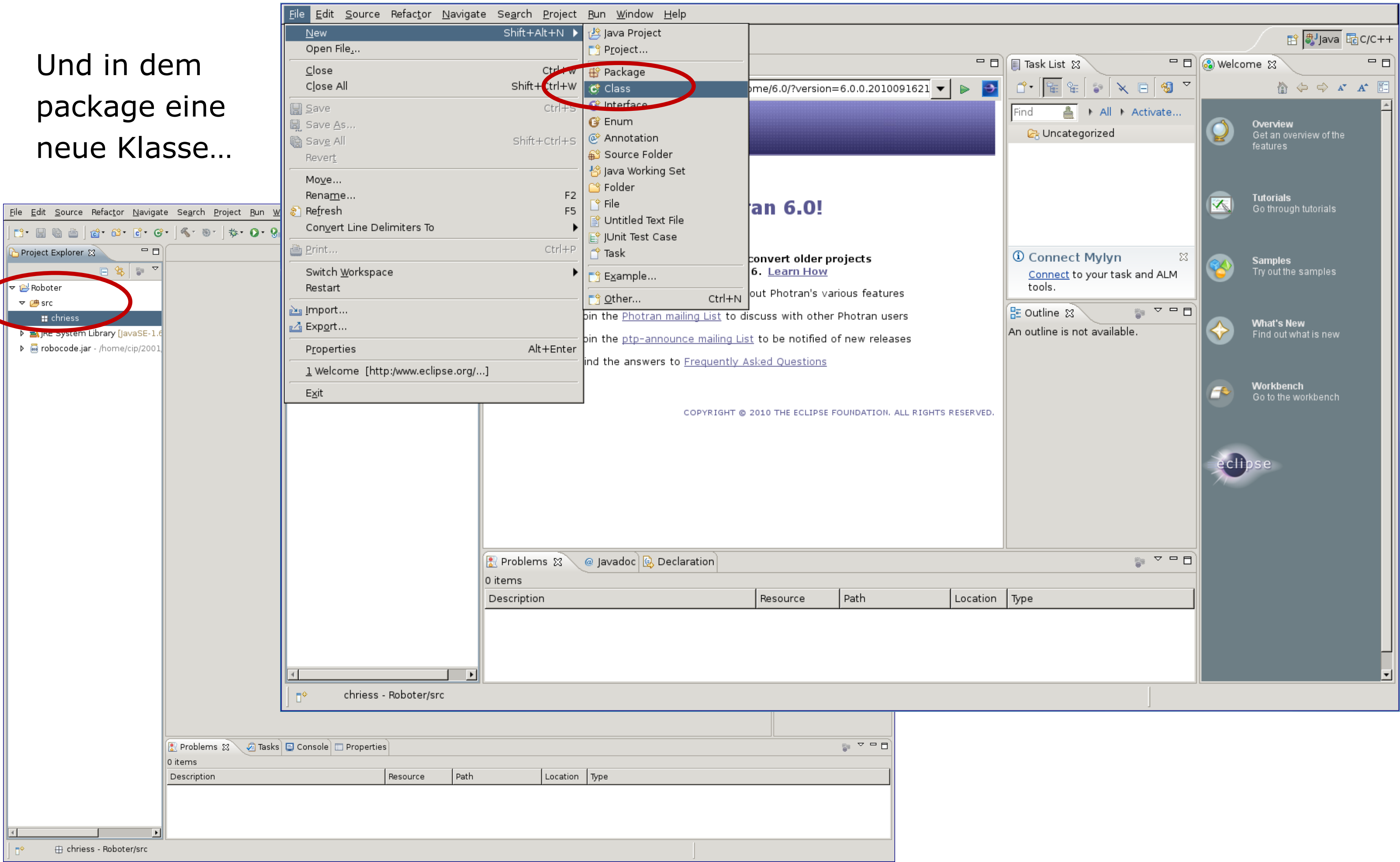
Connect to your task and ALM tools.

An outline is not available.

0 items

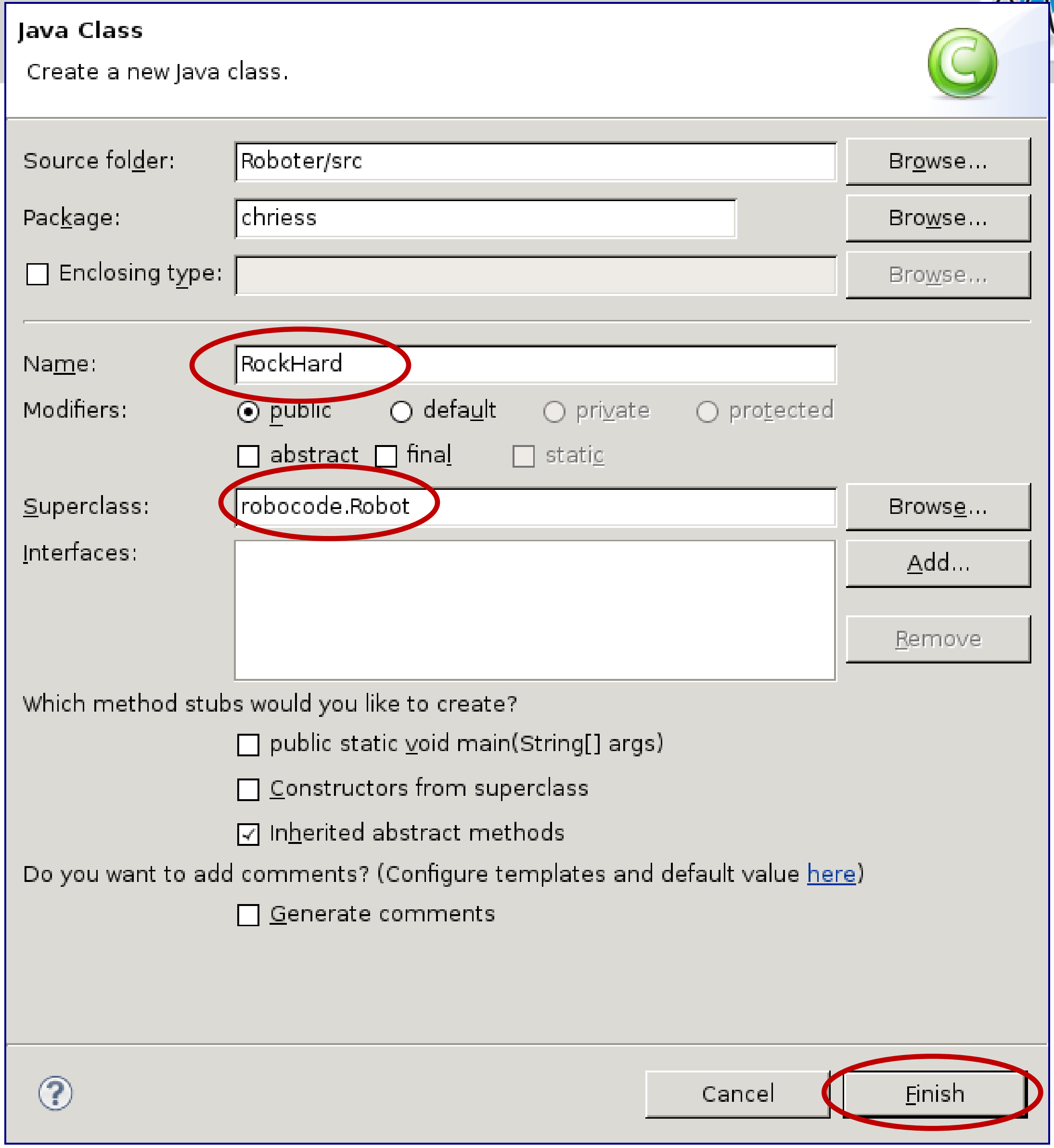
Step by step

- Und in dem package eine neue Klasse...



Step by step

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



Java Class
Create a new Java class.

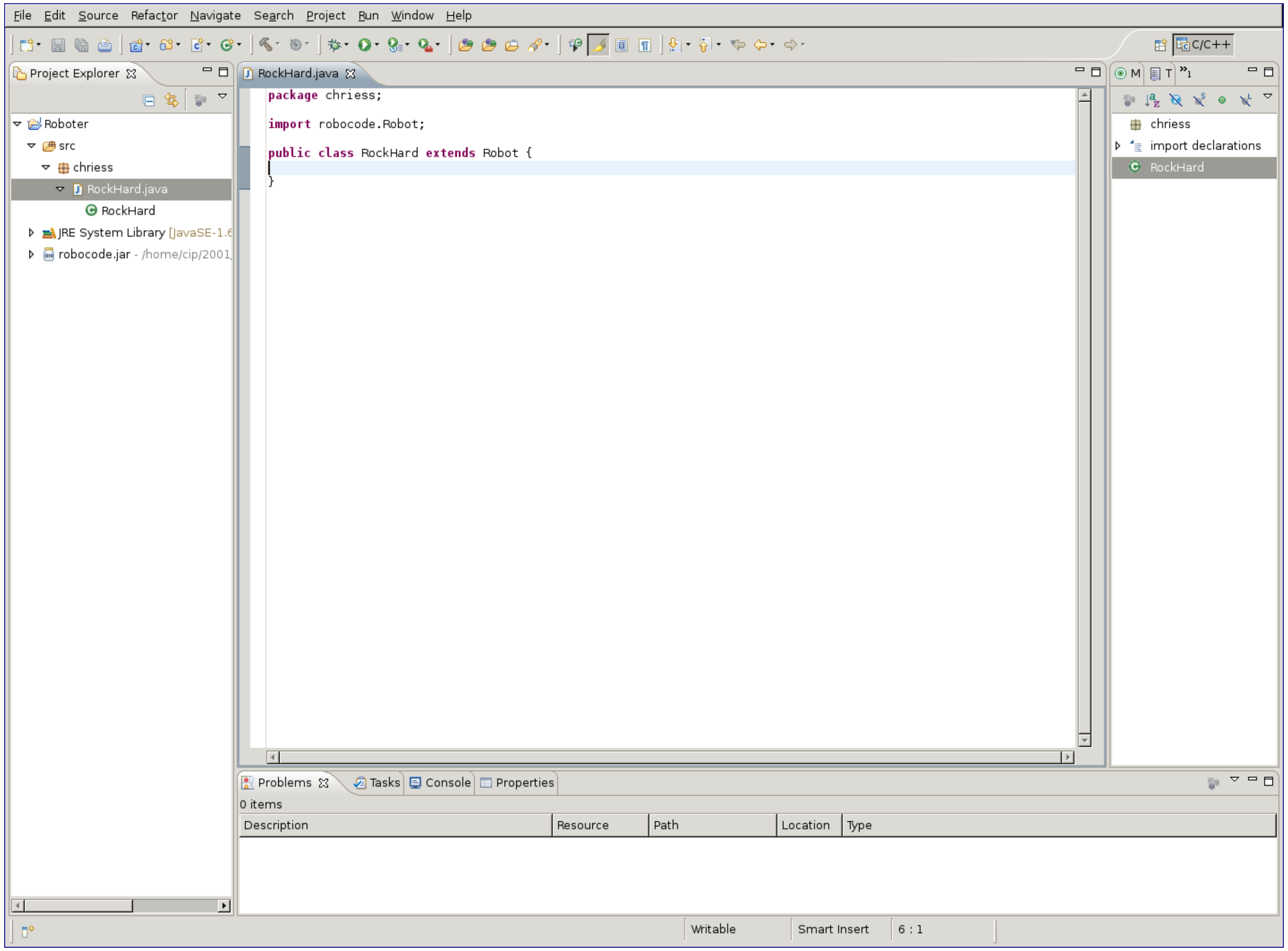
Source folder: Roboter/src
Package: chriess
☐ Enclosing type:
Name: RockHard
Modifiers: [✓ public] default private protected
☐ abstract ☐ final ☐ static
Superclass: robocode.Robot
Interfaces:

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!

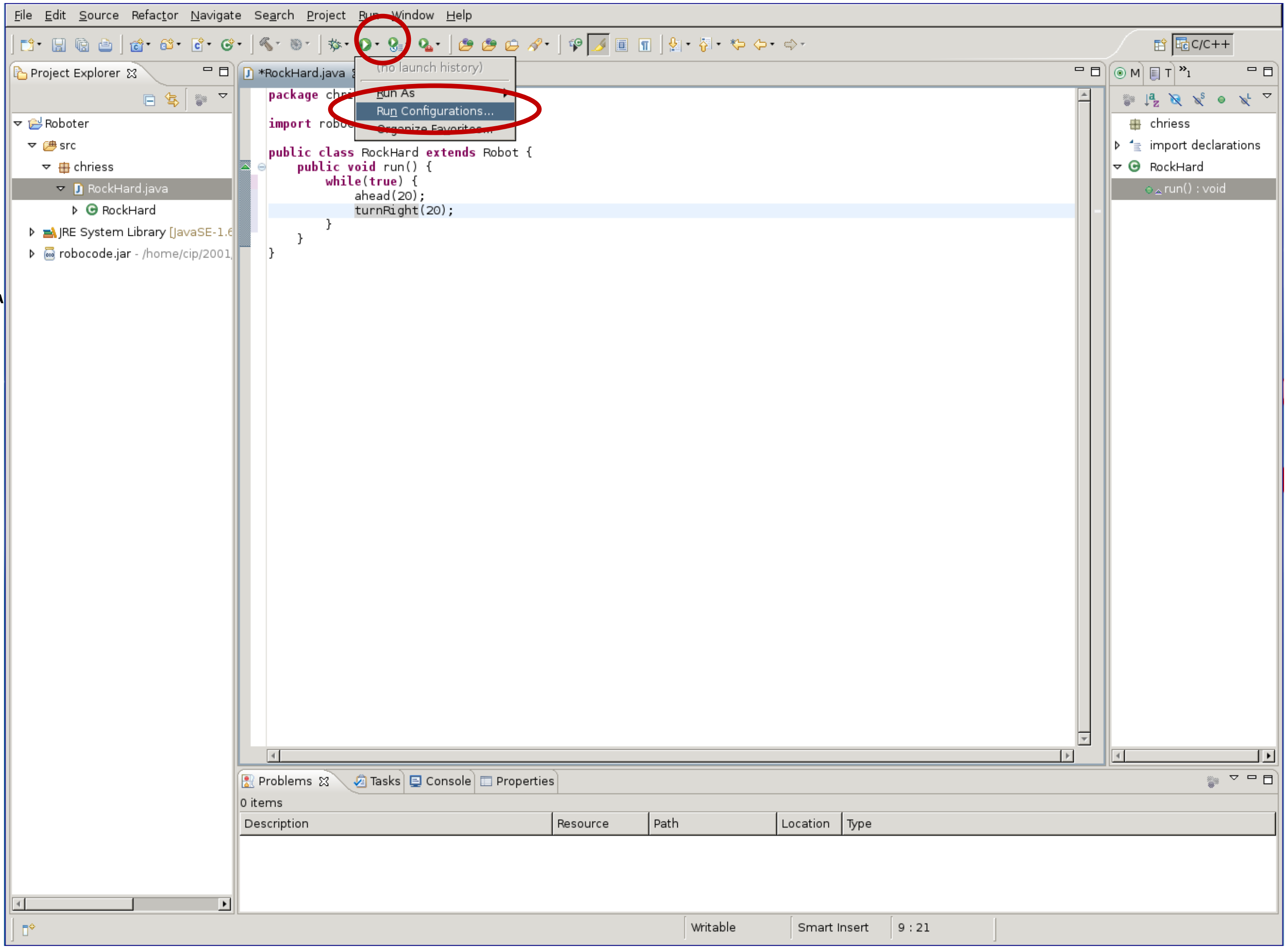
- …zum Schluss muss man noch einstellen, wie das Programm ausgeführt wird (siehe nächste Folie)



```
package chriess;

import robocode.Robot;

public class RockHard extends Robot {

}
```

Step by step

- Konfiguration der Laufzeit-umbegung: „Run Configurations"



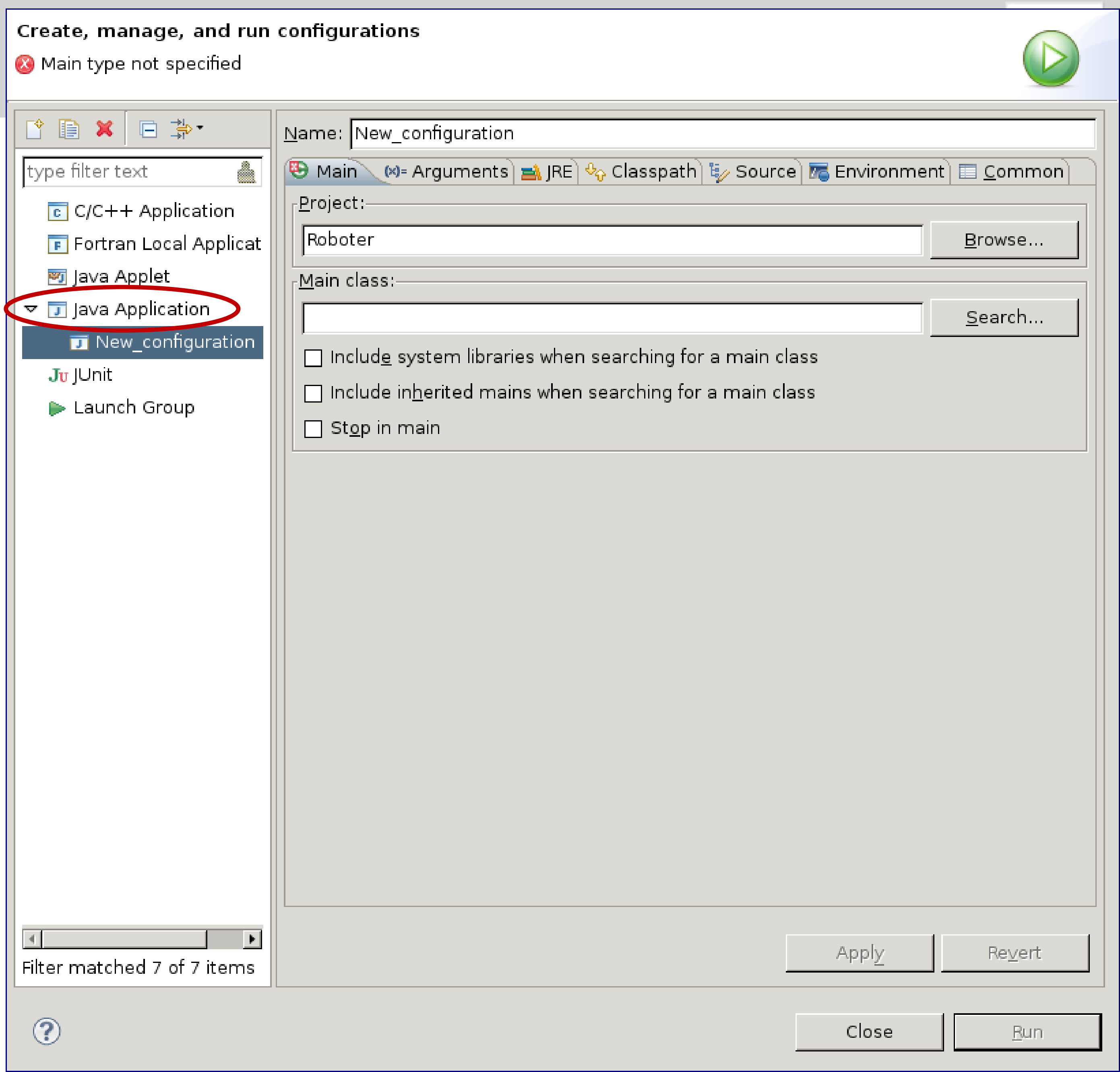
```
package chri
import robo

public class RockHard extends Robot {
    public void run() {
        while(true) {
            ahead(20);
            turnRight(20);
        }
    }
}
```

0 items

Description	Resource	Path	Location	Type

Step by step

- Doppelklick auf „Java Application" erzeugt eine neue Konfiguration



Create, manage, and run configurations

Main type not specified

Name: New_configuration

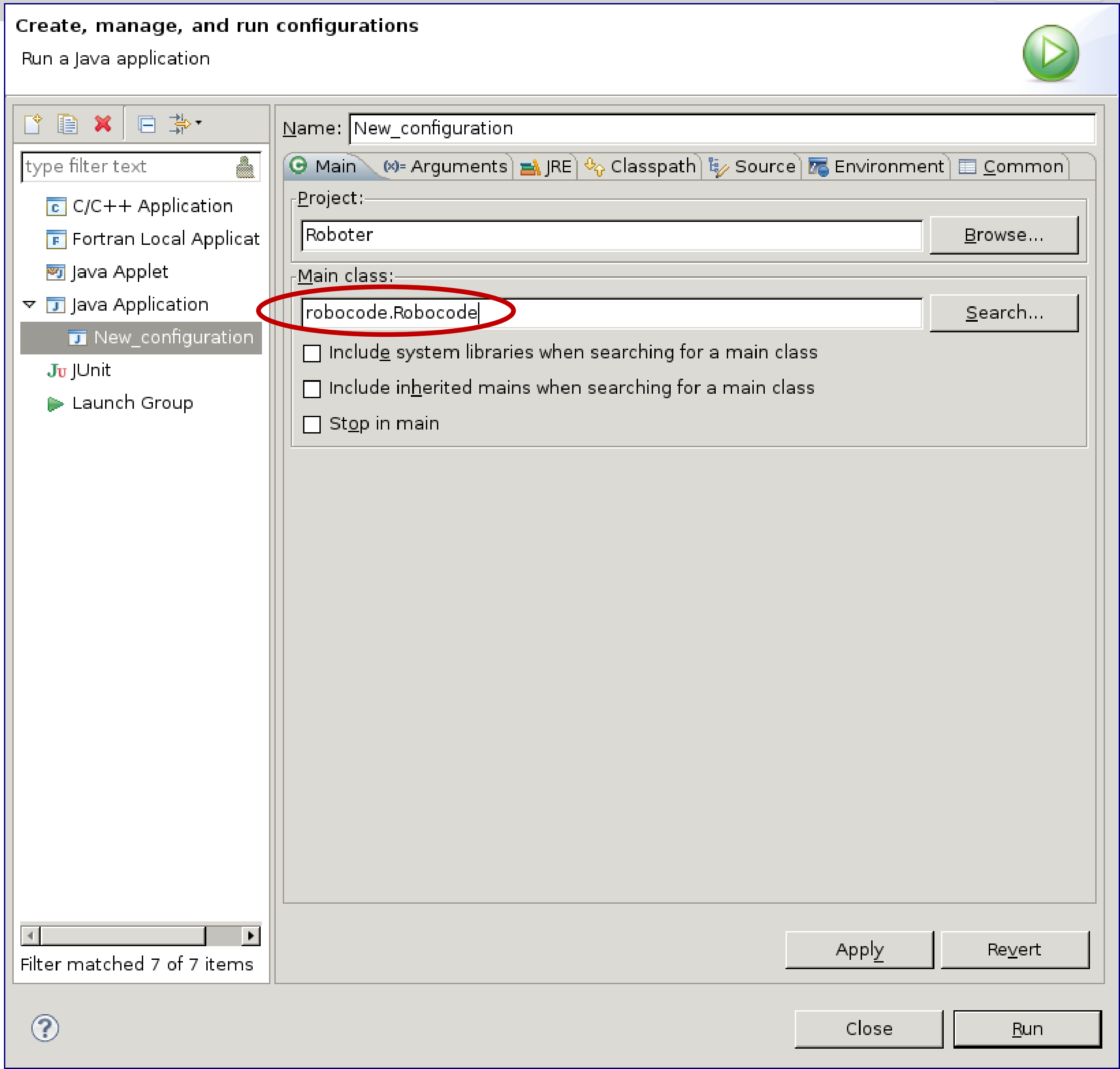
Project: Roboter

Main class:

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

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

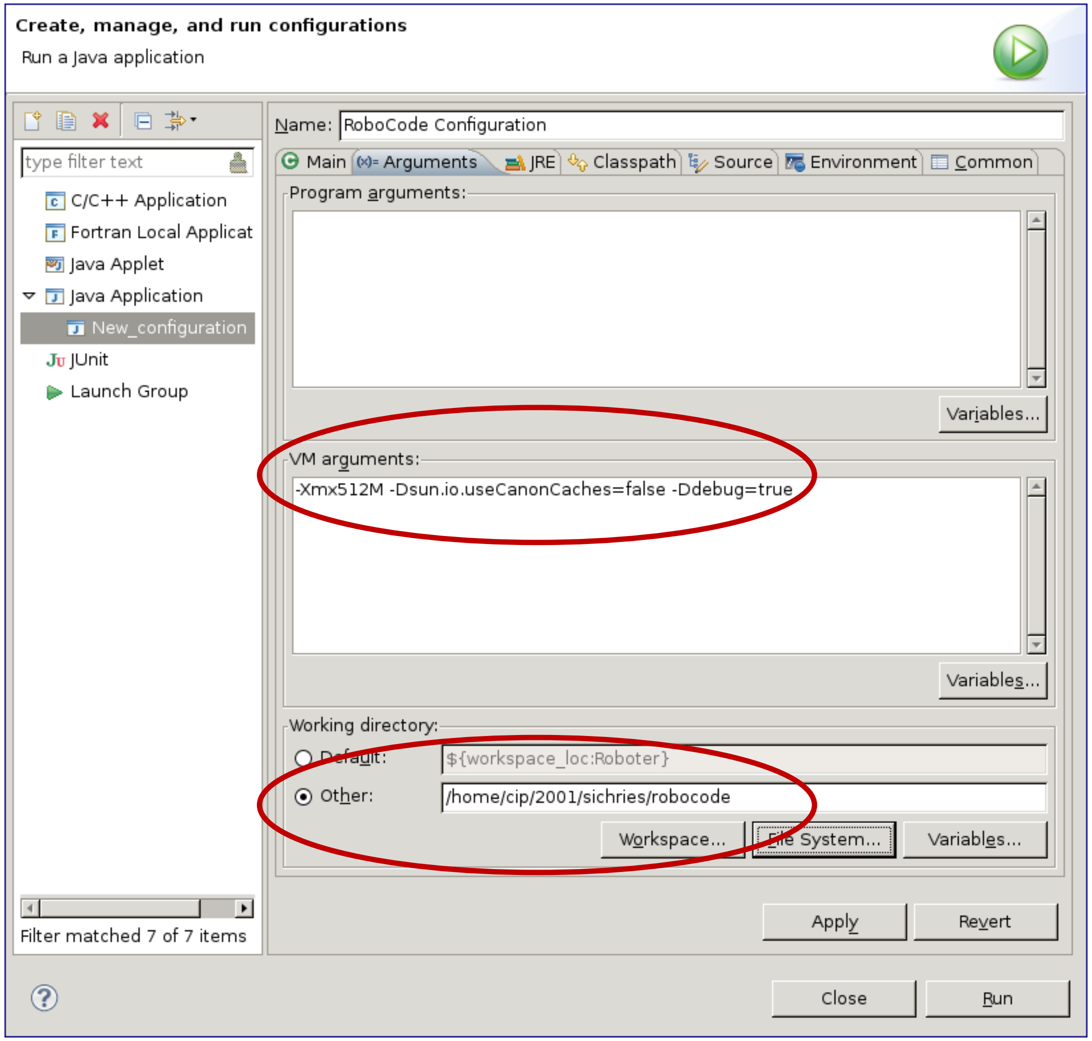
Project: Roboter

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

Step by step

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


Create, manage, and run configurations

Run a Java application

Name: RoboCode Configuration

Program arguments:

VM arguments:

```
-Xmx512M -Dsun.io.useCanonCaches=false -Ddebug=true
```

Working directory:

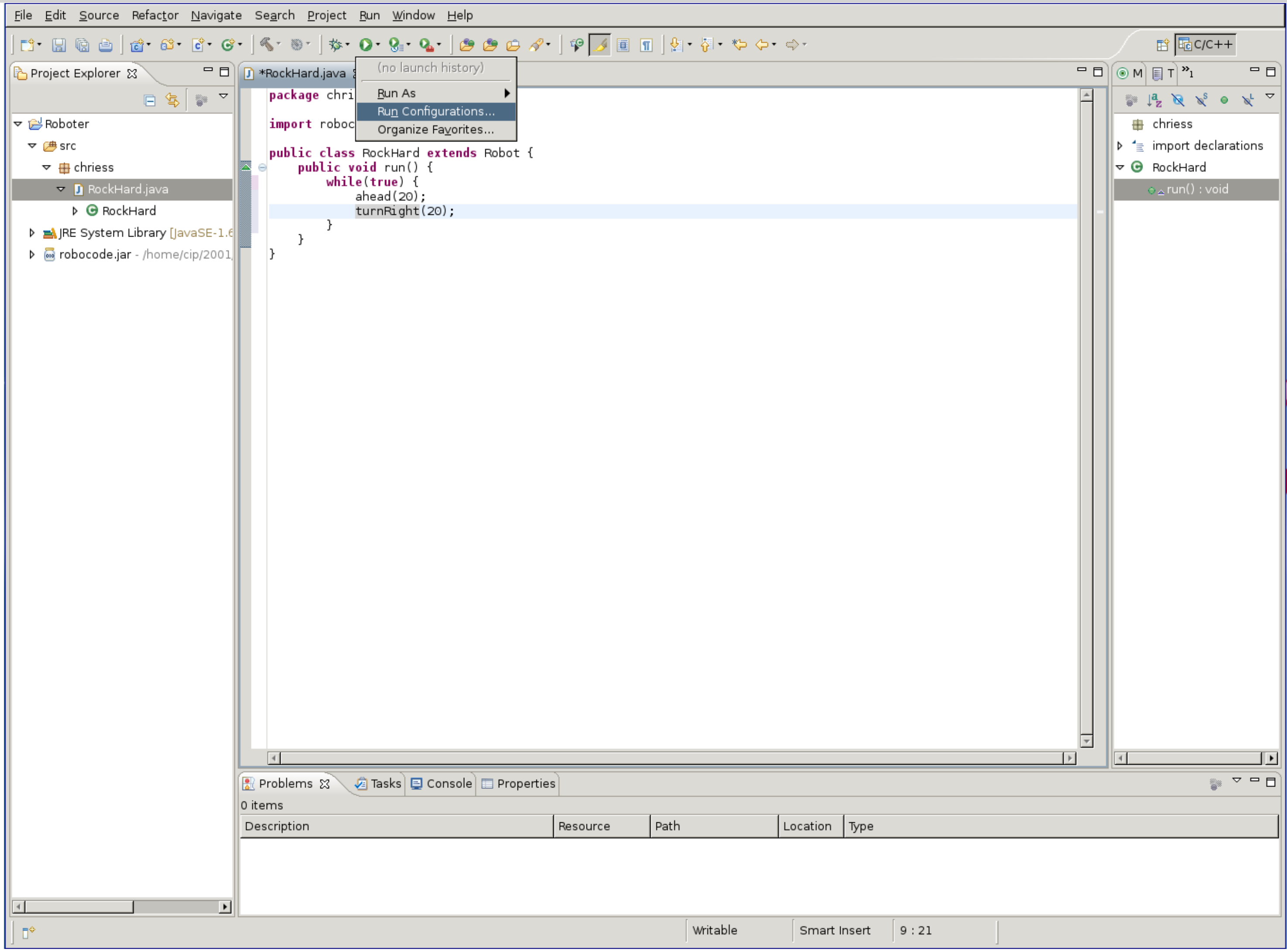
[ ] Default: ${workspace_loc:Roboter}

[✓ Other:] /home/cip/2001/sichries/robocode

Filter matched 7 of 7 items

Step by step

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



```
package chri
import roboc
public class RockHard extends Robot {
    public void run() {
        while(true) {
            ahead(20);
            turnRight(20);
        }
    }
}
```

0 items

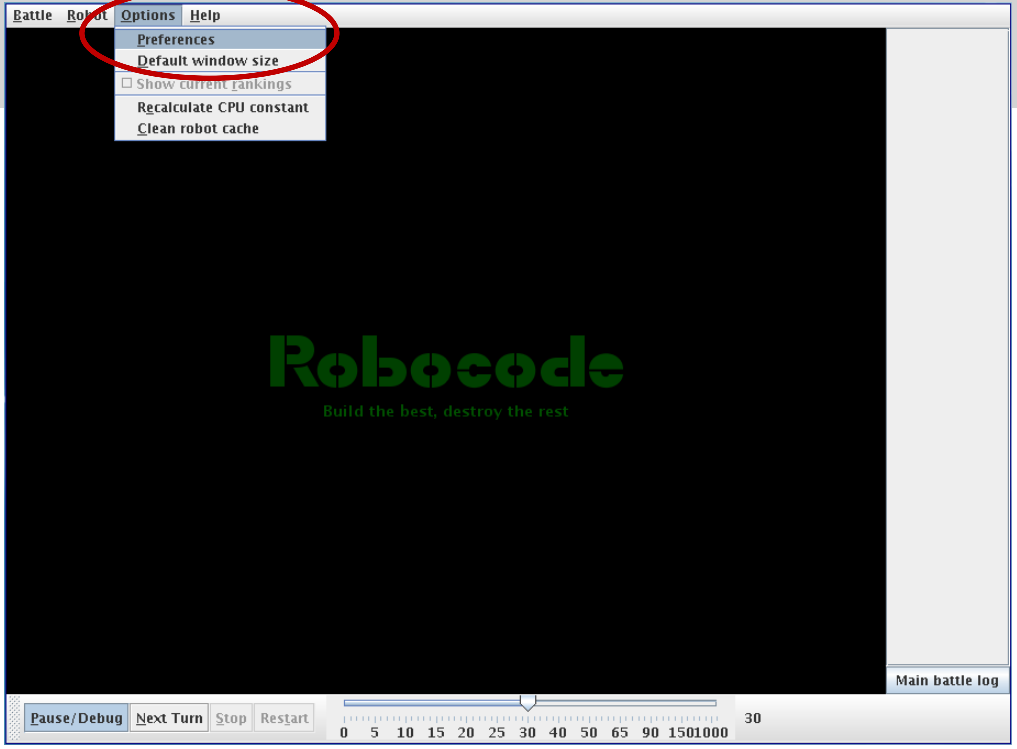
Description	Resource	Path	Location	Type

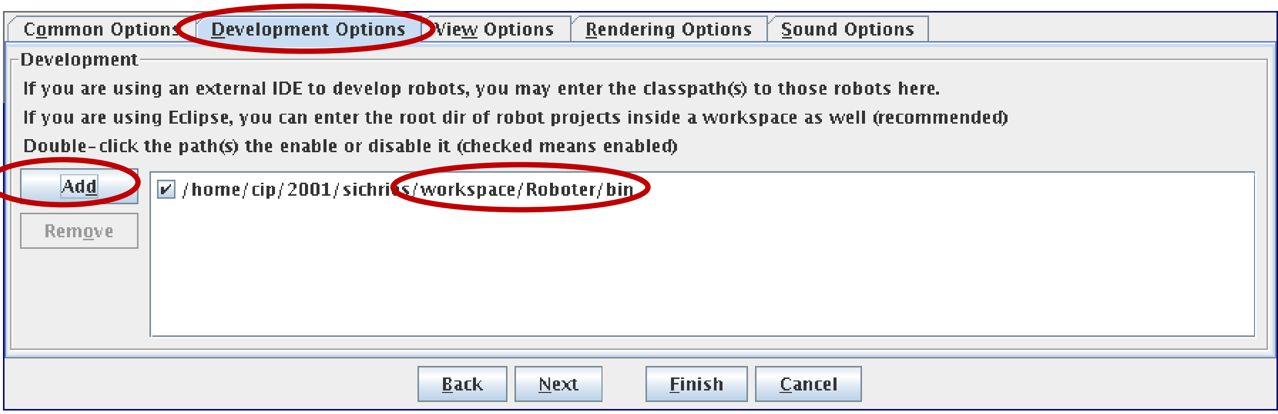
Writable | Smart Insert | 9 : 21

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`**





Development

If you are using an external IDE to develop robots, you may enter the classpath(s) to those robots here.

If you are using Eclipse, you can enter the root dir of robot projects inside a workspace as well (recommended)

Double-click the path(s) the enable or disable it (checked means enabled)

☑ /home/cip/2001/sichrils/workspace/Roboter/bin