

Fakultät für Informatik, Institut für Robotik

JAVA mit NXT

Programmierung mit Methoden und Objekten

Ute Ihme





DAS SPIELFELD: Legostadt

Aufgabe 1: Robotersteuerung

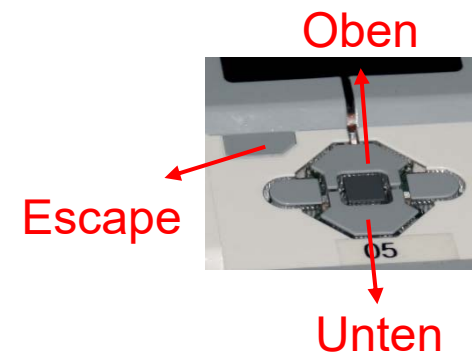
Start: Startfeld

Ende:

Button oben: Parkplatz am Haus

Button unten: Parkplatz am Krankenhaus

Button Escape: Programmende



Der Roboter soll vom Startfeld entweder zum Haus (Button oben) oder zum Krankenhaus (Button unten) fahren. Wird keine der Optionen ausgewählt, so soll eine erneute Eingabe erfolgen. Mit der Escape-Taste soll das Programm abgebrochen werden.

- Fortsetzung nächste Folie -



DAS SPIELFELD: Legostadt

Abfrage von NXT Buttons

Warten auf Knopfdruck:

```
Button.waitForAnyPress( );
```

Abfrage, ob Knopf links gedrückt ist:

```
Button.LEFT.isDown()
```



DAS SPIELFELD: Legostadt

Aufgabe 1: Robotersteuerung

Lösungsweg:

1. Erstellen Sie eine Klasse, die die Main-Methode erstellen!

- Fortsetzung nächste Folie -

1. Erstellen Sie eine Klasse, die die Main-Methode erstellen!

The screenshot shows the Eclipse IDE interface. On the left, the 'Paket-Explorer' (Package Explorer) shows a project structure with a package named 'BeispielObjekt' containing a sub-package 'src' and a file 'Steuerung.java'. The main editor window displays the code for 'Steuerung.java':

```
1 public class Steuerung {  
2  
3  
4     public static void main(String[] args) {  
5         // TODO Automatisch generierter  
6  
7     }  
8  
9 }  
10
```

A red arrow points from the 'Neue Java-Klasse' dialog box to the 'main' method in the code editor. The dialog box is titled 'Neue Java-Klasse' and contains the following fields and options:

- Quellenordner: BeispielObjekt/src
- Paket: (Standard)
- Übergeordneter Typ: (Standard)
- Name: Steuerung
- Modifizierer: Öffentlich (public), Standard (default), Privat (private), Geschützt (protected), Abstrakt (abstract), Final (final), Statisch (static)
- Superklasse: java.lang.Object
- Schnittstellen: (Empty list)
- Welche Methoden-Stubs sollen erstellt werden?
 - public static void main(String[] args)
 - Konstruktoren aus Superklasse
 - Übernommene abstrakte Methoden
- Möchten Sie Kommentare hinzufügen? (Vorlagen und Standardwert [hier](#) konfigurieren)
 - Kommentare generieren

Buttons at the bottom: Fertigstellen, Abbrechen.



DAS SPIELFELD: Legostadt

Aufgabe 1: Robotersteuerung

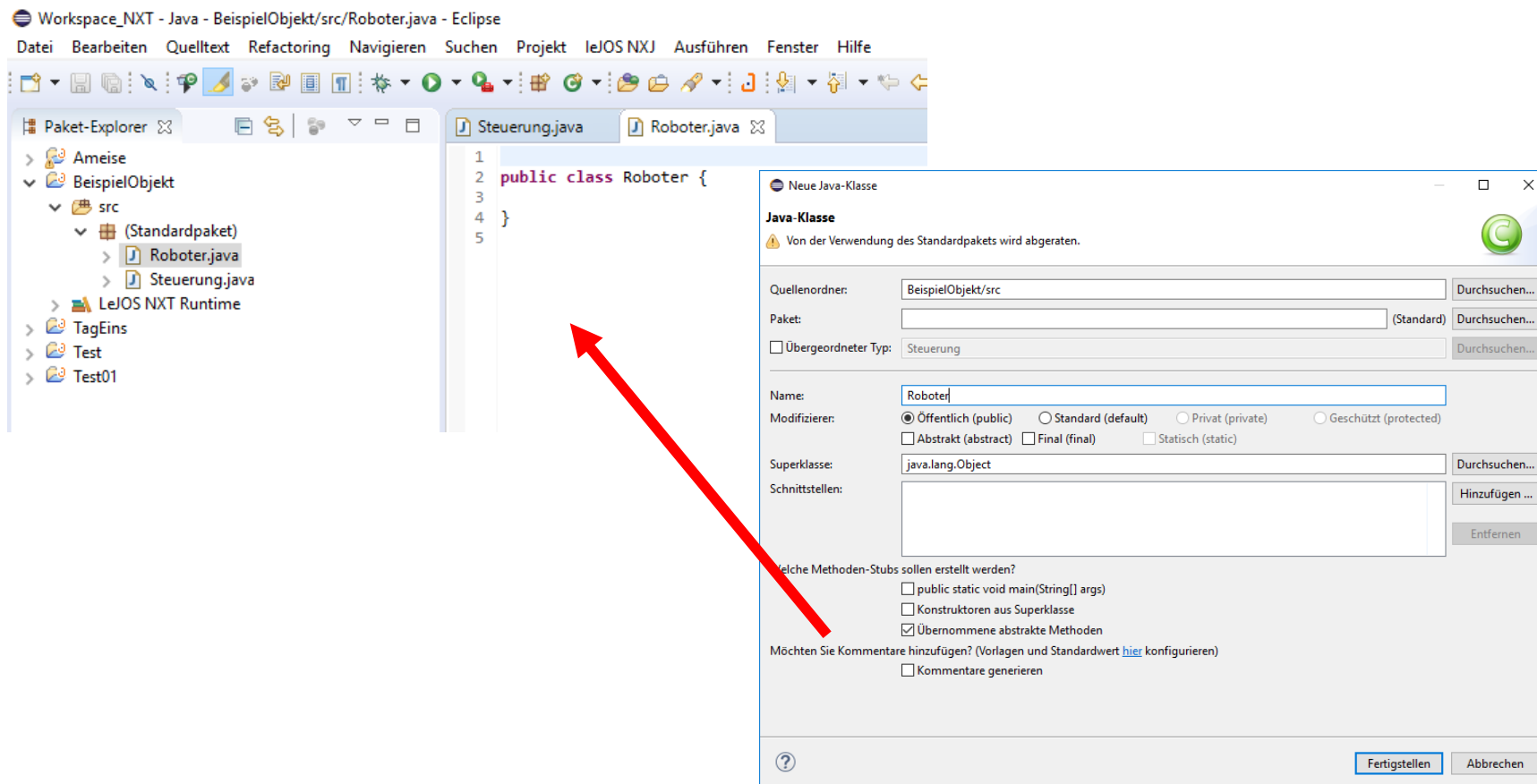
Lösungsweg:

1. Erstellen Sie eine Klasse, die die Main-Methode erstellen!
2. Erstellen Sie eine weitere Klasse **Roboter**, die die Methoden zur **Robotersteuerung** enthalten soll!

- Fortsetzung nächste Folie -



2. Erstellen Sie eine weitere Klasse Roboter, die die Methoden zur Robotersteuerung enthalten soll!



The screenshot shows the Eclipse IDE interface. On the left, the 'Paket-Explorer' (Package Explorer) shows a project structure with a package named 'BeispielObjekt' containing a sub-package 'src'. Inside 'src', there are two files: 'Roboter.java' and 'Steuerung.java'. The 'Roboter.java' file is selected, and its content is visible in the editor:

```
1  
2 public class Roboter {  
3  
4 }  
5
```

On the right, the 'Neue Java-Klasse' (New Java Class) dialog is open. The 'Name' field contains 'Roboter'. The 'Modifizierer' (Modifiers) section has 'Öffentlich (public)' selected. The 'Superklasse' (Superclass) is set to 'java.lang.Object'. The 'Schnittstellen' (Interfaces) section is empty. The 'Welche Methoden-Stubs sollen erstellt werden?' (Which method stubs should be created?) section has 'Übernommene abstrakte Methoden' (Overridden abstract methods) checked. The 'Möchten Sie Kommentare hinzufügen?' (Do you want to add comments?) section has 'Kommentare generieren' (Generate comments) unchecked. The 'Fertigstellen' (Finish) button is highlighted.

A red arrow points from the 'Roboter.java' file in the Package Explorer to the 'Name' field in the 'Neue Java-Klasse' dialog.



DAS SPIELFELD: Legostadt

Aufgabe 1: Robotersteuerung

Lösungsweg:

3. Schreiben Sie in der Klasse Roboter jeweils eine Methode für
 - Vorwärts fahren
 - Rückwärts fahren
 - Links fahren
 - Rechts fahren und
 - Anhalten
 - Die Zeit, die der Roboter fahren soll, soll als Parameter übergeben werden

- Fortsetzung nächste Folie -



3. Erstellen von Methoden

```
public static <<RUECKGABETYP>> <<METHODENNAME>> (<<PARAMETERLISTE>>){  
    // Methoden-Rumpf: hier den  
    // auszufuehrenden Code einfuegen  
}
```

<<RUECKGABETYP>>:

- **void:** kein Rückgabetyt
- **integer, double, float:** entsprechende Datentypen

<<METHODENNAME>>: Name der Methode

<<PARAMETERLISTE>>: Liste der Parameter, die an die Methode übergeben werden.



3. Erstellen von Methoden

Beispiel Deklaration in einer weiteren Klasse

```
import lejos.nxt.Motor;  
import lejos.util.Delay;  
  
public class Roboter {  
  
    public static void vorwaerts(int zeit){  
        Motor.B.forward();  
        Motor.C.forward();  
        Delay.msDelay(zeit);  
    }  
}
```



3. Erstellen von Methoden

Beispiel Aufruf aus der Hauptklasse

```
public class Steuerung {  
  
    public static void main(String[] args) {  
        Roboter.vorwaerts(500);  
    }  
}
```



3. Erstellen von Methoden

Beispiel Deklaration und Aufruf in der Hauptklasse

```
public class Steuerung {  
  
    public static void main(String[] args) {  
        rueckwaerts(500);  
    }  
  
    public static void rueckwaerts (int zeit){  
  
        Motor.B.backward();  
        Motor.C.backward();  
        Delay.msDelay(zeit);  
  
    }  
}
```



3. Erstellen von Methoden

Erstellen von Methoden zur Arbeit mit Objekten

- Erstellung einer Klasse
- Bei den Methoden wird das Wort `static` weggelassen

```
public class Roboter {  
  
    public void vorwaerts(int zeit){  
        Motor.B.forward();  
        Motor.C.forward();  
        Delay.msDelay(zeit);  
    }  
  
}
```



3. Erstellen von Methoden

Erstellen von Objekten und Aufrufen ihrer Methoden
- Realisiert in main-Methode in der Hauptklasse

```
public class Steuerung {  
  
    public static void main(String[] args) {  
  
        //Deklaration eines Objektes  
        Roboter roboter = new Roboter();  
  
        // Aufruf der Methode vorwaerts  
        roboter.vorwaerts(500);  
    }  
}
```



DAS SPIELFELD: Legostadt

Aufgabe 1: Robotersteuerung

Lösungsweg:

4. Schreiben Sie in Ihrer Hauptklasse jeweils eine Methode für die die Fahrt zum Krankenhaus und für die Fahrt zum Haus.
5. Realisieren Sie unter Verwendung einer eigenen Methode in der Hauptklasse die gesamte Aufgabenstellung. Verwenden Sie while-Schleifen und if-else-Abfragen.