hochschule mannheim

Fakultät für Informatik, Institut für Robotik

EV3 - Laborpraktikum II
Programmierung mit JAVA

Ute Ihme

Wiederholung von Laborpraktikum I

Ausführliche Informationen
Folien zum Laborpraktikum I

Bildschirmanzeigen

1. Nutzung des Standard JAVA Befehls

```
System.out.println("Hello World");
```

2. Nutzung des lejos Befehls
   Für nachfolgende Befehle ist folgender import Befehl notwendig

```
import lejos.hardware.lcd.LCD;
```

Anzeige von Strings:

```
LCD.drawString(String, Spalte, Zeile);
```

Anzeige von Zahlen:

```
LCD.drawInt(zahl, Spalte, Zeile);
```

Bildschirmlöschen:

```
LCD.clearDisplay();
```

Pausenbefehle

1. Warten darauf, dass ein Knopf des EV3 Steins gedrückt wird

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

Für die Nutzung dieses lejos Befehls wird die import-Funktion benötigt:

```
import lejos.hardware.Button;
```

2. Nutzung eines leJos Pausen – Befehls: msDelay

```
Delay.msDelay(1000);
```

Für die Nutzung dieses lejos Befehls wird die import-Funktion benötigt:

```
import lejos.utility.Delay;
```

DAS SPIELFELD: Legostadt

Steuerung zweier Motoren mittels Zeitangaben

Setzen einer definierten Geschwindigkeit:

```
Motor.B.setSpeed(400);
Motor.C.setSpeed(400);
```

Vorwärtsfahren:

Motor.***B.forward();***
Motor.***C.forward();***

Rückwärtsfahren:

Motor.***B.backward();***
Motor.***C.backward();***

Anhalten mit Bremsen:

Motor.***B.stop();***
Motor.***C.stop();***

Für die Nutzung der Motor-Befehle wird die import-Funktion benötigt:

```
import lejos.hardware.motor.Motor;
```

Überprüfen Sie, dass die Motoren in den Ports B und C angeschlossen sind! Wenn nicht, dann die Stecker in die entsprechenden Ports stecken oder Portangabe im Programm entsprechend ändern!

Laborpraktikum II

Hinweise zur Bearbeitung der Praktikumsaufgaben

- Das Praktikum wird am Spielfeld Legostadt durchgeführt
- Jede Aufgabe ist eine eigenständige Aufgabe.
- Setzen Sie die in den Aufgaben formulierten Anweisungen unbedingt um.
  Nichtbeachtung führt zu Punktabzug!
- Für die Abnahme gilt folgende Verfahrensweise:
  - Zeigen der Aufgabe mittels Roboter
  - Zeigen des Quellcodes

DAS SPIELFELD: Legostadt

Aufgabe 1: Robotersteuerung

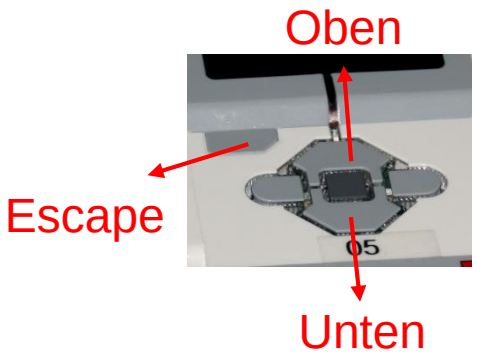
Start: Startfeld

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 EV3 Buttons

Warten auf Knopfdruck:

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

Abfrage, ob Knopf oben gedrückt ist:

```
Button.getButtons()==Button.ID_UP
```

Löschen des Buttonabfrageergebnisses

```
Button.discardEvents();
```

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 -

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 -

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.

DAS SPIELFELD: Legostadt

Aufgabe 1: Robotersteuerung

Start: Startfeld

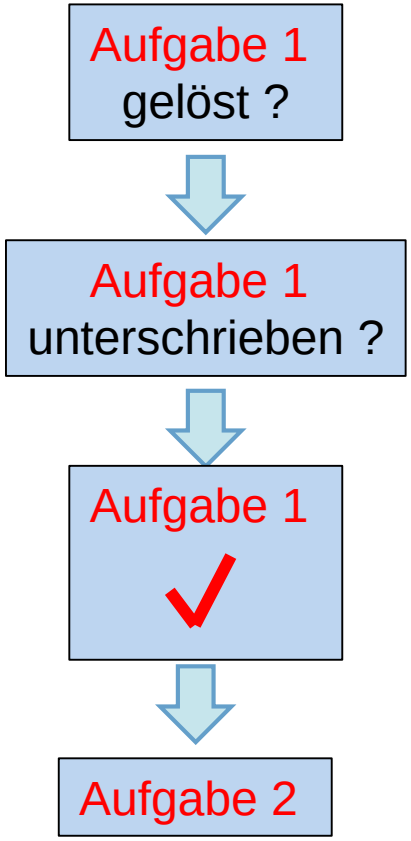
Ende:

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.



DAS SPIELFELD: Legostadt

Aufgabe 2: Anhalten mittels Tastsensor

Start: Startfeld

Ende: Parkplatz Berghütte

Der Roboter soll von der Startfliese zum Parkplatz Berghütte fahren. Dabei soll der Roboter autonom einparken, dass heißt, sobald der Tastsensor des Roboters die Wand am Parkplatz berührt soll der Roboter anhalten.

Die Aufgabe soll als eigenständige Methode in der Hauptklasse realisiert werden. Initialisierung und Sensorabfragen sollen innerhalb dieser Methode realisiert werden.

Schließen Sie den Tastsensor an Port 1 an!

DAS SPIELFELD: Legostadt

Zur Arbeit mit Sensoren

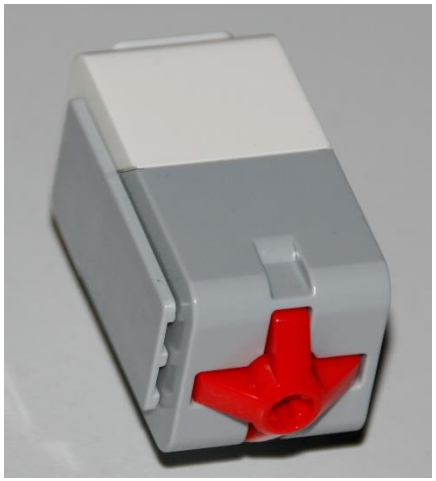
```
import lejos.hardware.port.SensorPort;
import lejos.hardware.sensor.*;
import lejos.robotics.*;
```

Arbeitsanweisung:
Fügen Sie **jetzt** diese drei Zeilen in ihr Programm an den Anfang, wo alle anderen import Funktionen stehen ein.

Hinweis:
Die Initialisierung der Sensoren und die Abfrage der Messwerte erfolgt bei allen Sensoren nach dem gleichen Prinzip.
Wichtig ist, dass stets der Port (S1, S2, S3 bzw. S4) in der Initialisierung verwendet wird, an dem der Sensor tatsächlich angeschlossen ist.

DAS SPIELFELD: Legostadt

Berührungssensor / Tastsensor



- Abfrage, ob Sensor gedrückt
- Werte des Sensors
  - 0: Sensor nicht gedrückt
  - 1: Sensor gedrückt

- Schließen Sie den Tastsensor an Port 1 an!

DAS SPIELFELD: Legostadt

Berührungssensor / Tastsensor

Initialisierung:

```
SensorModes sensor1 = new EV3TouchSensor(SensorPort.S1);
SampleProvider touch = sensor1.getMode("Touch");
```

Jeweiligen Anschlußport angeben (S1, S2, S3 oder S4)

Abfrage der Messwerte:

```
// Initialisierung der Messwerte
float pressed =0;
float sample[] = new float[touch1.sampleSize()];

// Abfrage der Sensorwerte
touch1.fetchSample(sample, 0);
pressed = sample[0];
```

Die Variable **pressed** enthält die Information, über den Zustand des Tastsensors. Diese gilt es im Programm abzufragen.

DAS SPIELFELD: Legostadt

Beispielprogramm: Tastsenor

Im Beispiel ist der Tastsensor am Port 1.

```
    LCD.drawString("Tastsensor druecken", 0, 1);
    while(pressed==0){
        // Abfrage Tastsensor
        touch1.fetchSample(sample, 0);
        pressed =sample[0];
        zahl=zahl+1;
    }
    LCD.drawString("zahl =", 0, 4);
    LCD.drawInt(zahl, 0, 3);
    Delay.msDelay(2000);

}
```

Das Programm erhöht eine Variable um 1, bis der Tastsensor gedrückt wird und zeigt anschließend das Ergebnis an.

DAS SPIELFELD: Legostadt

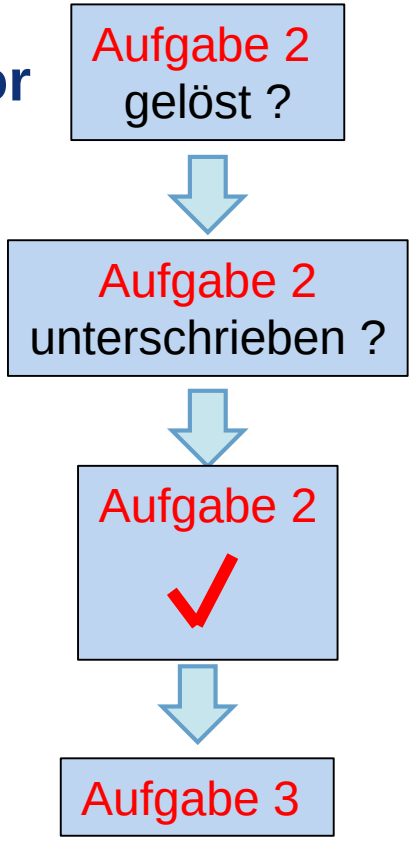
Aufgabe 2: Anhalten mittels Tastsensor

Start: Startfeld

Ende: Parkplatz Parkplatz Berghütte

Der Roboter soll von der Startfliese zum Parkplatz Berghütte fahren. Dabei soll der Roboter autonom einparken, dass heißt, sobald der Tastsensor des Roboters die Wand am Parkplatz berührt soll der Roboter anhalten.

Die Aufgabe soll als eigenständige Methode in der Hauptklasse realisiert werden. Initialisierung und Sensorabfragen sollen innerhalb dieser Methode realisiert werden.



DAS SPIELFELD: Legostadt

Aufgabe 3:
Einparken mittels Ultraschallsensor

Start: Parkplatz Bahnhof

Ende: Parkplatz Flughafen

Der Roboter soll vom Parkplatz Bahnhof zum Parkplatz Flughafen fahren. Dabei soll der Roboter autonom in einer Distanz von ca. 5 cm der Wand anhalten.

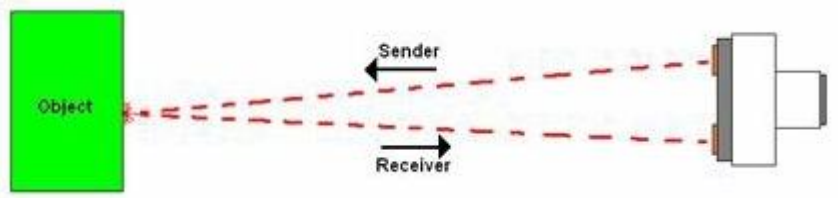
Die Aufgabe soll als eigenständige Methode in der Hauptklasse realisiert werden. Initialisierung und Sensorabfragen sollen innerhalb dieser Methode realisiert werden.

Ultraschallsensor an Port 4 anschließen!

DAS SPIELFELD: Legostadt

Ultraschallsensor

- Sensor sendet Ultraschall aus
- Schall wird von Hindernis reflektiert
- Reflektierter Schall wird vom Empfänger registriert
- Aus Laufzeit des Schalls kann auf die Entfernung geschlussfolgert werden
- Messbereich: 3 bis 250 cm
- Messgenauigkeit: +/- 1 cm
- Messwerte werden in Meter ausgegeben



DAS SPIELFELD: Legostadt

Ultraschallsensor

Initialisierung:

```
SensorModes sensor4 = new EV3UltrasonicSensor(SensorPort.S4);
SampleProvider us = sensor4.getMode("Distance");
```

Jeweiligen Anschlußport angeben (S1, S2, S3 oder S4)

Abfrage der Messwerte:

```
// Initialisierung der Messwerte
float distanz=10;
float sample[] = new float[us.sampleSize()];

// Abfrage der Messwerte
us.fetchSample(sample, 0);
distanz = sample[0];
```

DAS SPIELFELD: Legostadt

Beispiel: Ultraschallsensor

```
import lejos.hardware.lcd.LCD;
import lejos.hardware.port.SensorPort;
import lejos.hardware.sensor.*;
import lejos.robotics.*;
import lejos.utility.Delay;

public class UltraschallBeispiel {
   public static void main(String[] args) {

      // Inhalt nächste Folie
   }
}
```

Das Programm zeigt die Entfernung in Metern an, solange der Abstand größer ist als 10 cm.

DAS SPIELFELD: Legostadt

Beispiel: Ultraschallsensor

```
public static void main(String[] args) {

   // Initialisierung Ultraschallsensor
   SensorModes sensor4 = new EV3UltrasonicSensor(SensorPort.S4);
   SampleProvider schall1 = sensor4.getMode("Distance");

   // Initialisierung Messwerte
   float distanz=10;
   float sample[] = new float[schall1.sampleSize()];
```

Das Programm zeigt die Entfernung in Metern an, solange der Abstand größer ist als 10 cm.

DAS SPIELFELD: Legostadt

Beispiel: Ultraschallsensor

```
    while(distanz>=0.1){
        // Abfrage der Messwerte
        schall1.fetchSample(sample, 0);
        distanz = sample[0];
        // Anzeige Messwerte
        LCD.drawString("Weg: "+distanz, 0, 1);
        Delay.msDelay(100);
    }
}
```

Das Programm zeigt die Entfernung in Metern an, solange der Abstand größer ist als 10 cm.

DAS SPIELFELD: Legostadt

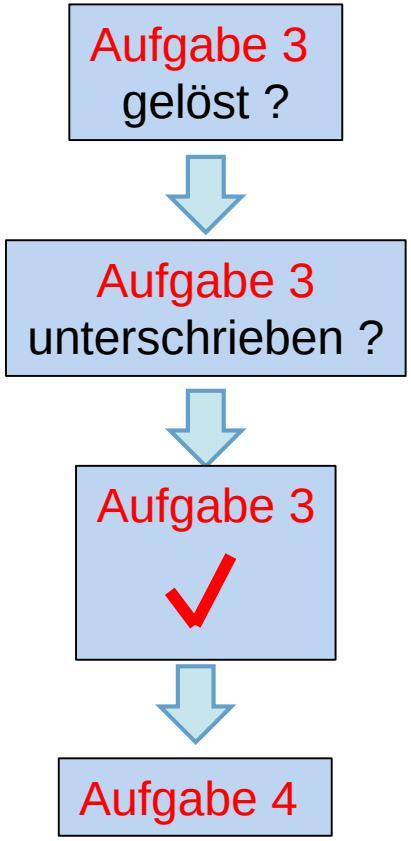
Aufgabe 3:
Einparken mittels Ultraschallsensor

Start: Parkplatz Bahnhof

Ende: Parkplatz Flughafen

Der Roboter soll vom Parkplatz Bahnhof zum Parkplatz Flughafen fahren. Dabei soll der Roboter autonom in einer Distanz von ca. 5 cm der Wand anhalten.

Die Aufgabe soll als eigenständige Methode in der Hauptklasse realisiert werden. Initialisierung und Sensorabfragen sollen innerhalb dieser Methode realisiert werden.



DAS SPIELFELD: Legostadt

Aufgabe 4: Einkaufsstraße / Farbsensor

Start: Parkplatz Flughafen

Ende: Farbfeld vor Laden

Farbsensor an Port 3 anschließen!

Der Roboter soll in Abhängigkeit von ermittelten Farbe am entsprechenden Ausflugsziel anhalten. Das Farbfeld wird über eine Zufallszahl ermittelt. Die Zufallszahl soll angezeigt werden. Verwenden Sie bei der Programmierung eine switch case Anweisung an geeigneter Stelle. Die Aufgabe ist in einer eigenen Methode zu realisieren.

0 – Gelb (Farb-ID: 3)

1 – Blau (Farb-ID: 2)

2 – Rot (Farb-ID: 0)

DAS SPIELFELD: Legostadt

Colorsensor – ColorID Mode

- Bestimmung der Farbe
- Jede Farbe hat einen Wert
- Werte für EV3 Colorsensor

Wert	Farbe
-1	keine
0	Rot
1	Grün
2	Blau
3	Gelb
4	Magenta
5	Orange
6	Weiß
7	Schwarz
8	Pink
9	Grau
10	Hellgrau
11	Dunkelgrau
12	Zyan
13	Braun

DAS SPIELFELD: Legostadt

Zur Verwendung des Farbsensors (ColorID Mode)

Initialisierung:

```
SensorModes colorSensor = new EV3ColorSensor(SensorPort.S3);
SampleProvider col = colorSensor.getMode("ColorID");
```

Jeweiligen Anschlußport angeben (S1, S2, S3 oder S4)

Abfrage der Messwerte:

```
// Intialisierung der Messwerte
int sampleSize = colorSensor.sampleSize();
float[] sample = new float[sampleSize];
int farbe;

// Abfrage der Messwerte
col.fetchSample(sample, 0);
// Umrechung float in integer
farbe = (int)sample[0];
```

Die Variable **farbe** gibt den erkannten Farbwert aus. Diese ist abzufragen.

DAS SPIELFELD: Legostadt

```
import lejos.hardware.Button;
import lejos.hardware.lcd.LCD;
import lejos.hardware.port.SensorPort;
import lejos.hardware.sensor.*;
import lejos.robotics.*;
import lejos.utility.Delay;

public class FarbsensorBeispiel {

   public static void main(String[] args) {

      // Inhalt nächste Folie
   }
}
```

Das Programm zeigt 4 Messwerte an.

DAS SPIELFELD: Legostadt

Beispielprogramm: Farbsensor

```
public static void main(String[] args) {

   // Initialisierung Farbsensor
   SensorModes colorSensor1 = new EV3ColorSensor(SensorPort.S3);
   SampleProvider col1 = colorSensor1.getMode("ColorID");

   // Intialisierung der Messwerte
   int SampleSize = colorSensor1.sampleSize();
   float[] sample = new float[SampleSize];

   // Variable für den Farbwert
   int farbe;
   LCD.clearDisplay();
```

Das Programm zeigt 4 Messwerte an.

DAS SPIELFELD: Legostadt

Beispielprogramm: Farbsensor

```
for(int i=1;i<=4;i++){
    LCD.drawString("Messung starten", 0, 1);
    LCD.drawString("Knopf druecken", 0, 2);
    Button.waitForAnyPress();
    // Messwert erfassen
    col1.fetchSample(sample, 0);
    // Umrechnung des Messwertes in eine Integervariable
    farbe = (int)sample[0];

    // Anzeige Messwert
    LCD.drawString("Farbwert:", 0, 3);
    LCD.drawInt(farbe, 0, 4);
    Delay.msDelay(2000);
    LCD.clearDisplay();
}
}
```

Das Programm zeigt 4 Messwerte an.

DAS SPIELFELD: Legostadt

Abfrage einer Zufallszahl

Benötigt wird die Import-Funktion:

```
import java.util.*;
```

Festlegung des Wertebereiches:

```
Random wuerfel = new Random();
```

Erzeugung einer Zufallszahl (integer) im Wertebereich 0…2:

```
int zahl zahl = wuerfel.nextInt(2);
```

DAS SPIELFELD: Legostadt

Aufgabe 4: Einkaufsstraße / Farbsensor

Start: Parkplatz Flughafen

Ende: Farbfeld vor Laden

Der Roboter soll in Abhängigkeit von ermittelten Farbe am entsprechenden Ausflugsziel anhalten. Das Farbfeld wird über eine Zufallszahl ermittelt. Die Zufallszahl soll angezeigt werden. (Verwendung von switch-case und eigene Methode)

0 – Gelb (Farb-ID: 3)

1 – Blau (Farb-ID: 2)

2 – Rot (Farb-ID: 0)