

Behaviour Konzept

```
package robots;
import lejos.nxt.*;
import lejos.robotics.subsumption.*;

public class Rescue {
    public static final NXTRegulatedMotor Motor_re = Motor.A;
    public static final NXTRegulatedMotor Motor_li = Motor.B;

    public static final LightSensor Licht_re = new LightSensor(SensorPort.S3);
```

Einfügen des Packages robots

Verwenden dieses Paketes

Klasse mit main() Methode

Definition von Motoren und Sensoren

Behaviour Konzept

```
public Rescue(){  
    Behavior b1 = new Vorwaerts();  
    Behavior b2 = new Rueckwaerts();  
    Behavior b3 = new Rechts();  
    Behavior b4 = new Links();  
    Behavior b5 = new Stoppe();  
    Behavior b6 = new Kreuzung();  
  
    Behavior[] bArray = {b1,b3,b4,b5};  
  
    Arbitrator schiri = new Arbitrator(bArray);  
    schiri.start();  
}
```

Definition von verschiedenen Behaviors (Verhalten)
Für jedes Behavior muss eine Klasse erstellt werden

Festlegen der aktiven Behaviours

Behaviour Konzept

```
public static void main(String[] args) {  
    // TODO Automatisch generierter Methodenstub
```

```
    int speed = 360;
```

Starten der Rescue Klasse

```
    new Rescue();
```

```
    Rescue.Motor_li.setSpeed(speed);
```

```
    Rescue.Motor_re.setSpeed(speed);
```

```
}
```

```
}
```

Klasse ohne Bedingungen II

```
package robots;
```

```
import lejos.robotics.subsumption.Behavior;
```

```
import lejos.util.Delay;
```

```
public class Vorwaerts implements Behavior {
```

```
    @Override
```

```
    public boolean takeControl() {
```

```
        // TODO Automatisch generierter Methodenstub
```

```
        return true;
```

```
    }
```

Klasse ohne Bedingungen II

```
@Override
public void action() {
    // TODO Automatisch generierter Methodenstub

    Rescue.Motor_li.backward();
    Rescue.Motor_re.backward();
}

@Override
public void suppress() {
    // TODO Automatisch generierter Methodenstub
}
}
```

Klasse mit Bedingungen I

```
package robots;

import lejos.robotics.subsumption.Behavior;
import lejos.util.Delay;

public class Rechts implements Behavior {

    @Override
    public boolean takeControl() {
        // TODO Automatisch generierter Methodenstub
        return Rescue.Licht_re.getLightValue()<40;
    }
}
```

Klasse mit Bedingungen II

```
@Override
public void action() {
    // TODO Automatisch generierter Methodenstub
    Rescue.Motor_Li.backward();
    Rescue.Motor_re.forward();
    Delay.msDelay(100);
    Rescue.Motor_Li.stop();
    Rescue.Motor_re.stop();
}
```

```
@Override
public void suppress() {
    // TODO Automatisch generierter Methodenstub
}
}
```

```
}
```