**Informationsblatt 1: Fahren -**  **LegoRobot mit zwei Motoren**

**Beispiel 1:**

Zuerst muss der Roboter im Code gebaut werden, in dem ein LegoRobot (EV3 Brick) und Motoren motA (der linke Motor) und motB (der rechte Motor) erzeugt werden. Die Motoren werden einzeln in Bewegung versetzt. Die Dauer der Bewegung wird mit Tools.delay() festgelegt.

**Programmcode**

|  |  |
| --- | --- |
| // MoveEx1.java  import ch.aplu.ev3.\*;  public class MoveEx1 {    MoveEx1()   {     LegoRobot robot = new LegoRobot();     Motor motA = new Motor(MotorPort.A);     Motor motB = new Motor(MotorPort.B);     robot.addPart(motA);     robot.addPart(motB);     motA.forward();     motB.forward();     Tools.delay(1000);     robot.exit();   }   public static void main(String[] args)   {     new MoveEx1();   } } |  |

**Erklärungen zum Programmcode:**

|  |  |
| --- | --- |
| import ch.aplu.ev3.\*. | Importiert die Klassenbibliothek Ev3JLibA |
| LegoRobot robot  =  new LegoRobot() | Erzeugt ein Roboterobjekt *robot* |
| Motor motA  =  new Motor(MotorPort.A) | Erzeugt den Motor *motA* (A entspricht Anschlussbuchse A) |
| Motor motB  =  new Motor(MotorPort.B) | Erzeugt den Motor *motB* (B entspricht Anschlussbuchse B) |
| robot.addPart(motA) | Fügt den Motor *motA* zum *robot* hinzu |
| Tools.delay(1000) | Bleibt während 1000 Millisekunden im aktuellen Zustand |
| robot.exit () | Beendet das Programm |
| MoveEx1 | Klassenname und Konstruktorname. Unter diesem Name wird das Programm auch auf dem EV3 gespeichert |
| public static void main(String[] args) | Jede Java-Applikation enthält eine Methode main(). Diese erstellt mit *new* eine Instanz des Programms, wodurch der Konstruktor ausgeführt wird |

**Beispiel 2:**

Dieser Roboter wird softwaremässig aus dem LegoRobot (Ev3-Brick) und aus einem Fahrwerk mit zwei synchronisierten Motoren zusammengebaut.

|  |  |
| --- | --- |
| Die Programme MoveEx2 und MoveEx2a bewirken dasselbe. Sie unterscheiden sich nur in der Methode forward(), die als **blockierende** oder **nicht blockierende** Methode verwendet werden kann. |  |

**Programmcode**

|  |  |
| --- | --- |
| // MoveEx2.java  import ch.aplu.ev3.\*;  class MoveEx2 {   MoveEx2()   {     LegoRobot robot = new LegoRobot();     Gear gear = new Gear();     robot.addPart(gear);     gear.forward(2000);     robot.exit();   }    public static void main(String[] args)   {     new MoveEx2();   } } | // MoveEx2a.java  import ch.aplu.ev3.\*;  class MoveEx2a {   MoveEx2a()   {     LegoRobot robot = new LegoRobot();     Gear gear = new Gear();     robot.addPart(gear);     gear.forward();     Tools.delay(2000);     robot.exit();   }   public static void main(String[] args)   {     new MoveEx2();  }  } |

**Erklärungen zum Programmcode:**

|  |  |
| --- | --- |
| Gear gear  =  new Gear() | Erzeugt ein Fahrwerk mit Motoren an den Anschlüssen A und B |
| robot.addPart(gear) | Das Fahrwerk wird zum *robot* hinzugefügt |
| forward(2000) | Der Roboter bewegt sich während 2000 ms vorwärts. Diese Methode forward ist **blockierend** und kehrt erst nach 2000 ms zurück |
| forward() | Methode *forward()* ohne Parameter im Programm rechts ist **nicht blockierend**. Der Roboter wird in den Forward-Zustand versetzt. und kann sofort auf weitere Befehle reagieren. Deaktiviert man die nächste Programmzeile (*Tools.delay(2000)*), so geht das Programm sofort zur nächsten Zeile (*robot.exit()*) und der Roboter hält an. Die nicht blockierenden Methoden werden insbesondere dann verwendet, wenn der Roboter während des Fahrens Sensorwerte zurückgeben muss |
| // Tools.delay(2000) | Wenn diese Zeile im Programm rechts aktiviert ist, bleibt der Roboter während 2000 ms in der Vorwärtsbewegung |

**Beispiel 3:**

Die einfachsten Programme erhalten Sie, wenn Sie den Standardroboter verwenden. TurtleRobot ist eine softwareseitige Erweiterung des EV3-Robots.  
  
Ein TurtleRobot fährt 200 Schritte vorwärts.

|  |  |
| --- | --- |
| // MoveEx3.java  import ch.aplu.ev3.\*;  class MoveEx3 {   MoveEx3()   {     TurtleRobot robot = new TurtleRobot();     robot.forward(200);     robot.exit();   }    public static void main(String[] args)   {     new MoveEx3();   } } |  |

**Erklärungen zum Programmcode:**

|  |  |
| --- | --- |
| TurtleRobot robot  =  newTurtleRobot() | Erzeugt einen voll fahrfähigen Roboter |
| forward(100) | Bewegt den Roboter 100 Schritte vorwärts. Diese Methode ist blockierend |
| robot.exit () | Die Kommunikation zwischen dem Roboter und dem Computer wird abgebrochen, Java-Programm wird beendet |