

Eine Einführung in die Objektorientierung mit Lego Mindstorms Robotern

Erfahrungsbericht aus dem Unterricht

Ruth Dietzel

Irmgardis-Gymnasium
Schillerstr. 98-102
D-50968 Köln
dietzel@royal.owl.de

Tim Rinkens

Winfriedstr. 39
D-33098 Paderborn
tau@upb.de

Abstract: Es werden die Erfahrungen aus einer in der Jahrgangsstufe 10 durchgeführten Unterrichtsreihe zur Einführung in die Objektorientierung mit Hilfe von Lego-Robotern vorgestellt. Zur Ansteuerung der Roboter entwickelte Tim Rinkens zwei Programme, die in dieser Reihe das erste Mal im Schuleinsatz getestet wurden. Aus Zeitgründen beschränkte sich die Einführung in die Konzepte der Objektorientierung auf die Begriffe "Objekt", "Klasse", "Vererbung" und "Ereignis". Bei den Lernenden konnte ein guter Lernerfolg erzielt werden.

1 Einleitung

Inzwischen wird an vielen Schulen im Informatikunterricht in das Konzept der Objektorientierung eingeführt. Dass die Arbeit mit realen Robotern für Kinder und Jugendliche Faszination und Motivation mit sich bringt, ist gut vorstellbar. Lässt sich Beides vielleicht für eine Unterrichtsreihe verknüpfen? Tim Rinkens entwickelte dazu zwei Programme:

- RCXDirectMode dient der direkten Steuerung des RCX-Bausteins, der das "Gehirn" der Lego Mindstorms Roboter darstellt.
- RCXDownload bietet eine Erleichterung im Umgang mit der Java Laufzeitumgebung leJOS. So können die Roboter auf einfache Art und Weise in der objektorientierten Programmiersprache Java programmiert werden.

Das wesentliche Ziel der Unterrichtsreihe bestand nun darin, die objektorientierte Sichtweise zu vermitteln, also in grundlegende Begriffe und Vorgehensweisen der Objektorientierung einzuführen. Nachgeordnet waren indessen die programmiertechnischen Details bzw. die Codierung der Klassen in Java. Vor dem Hintergrund der Vorkenntnisse der Lerngruppe und im Hinblick auf die Kürze der Unterrichtsreihe sollte der Schwerpunkt auf den Begriffen *Objekt*, *Klasse*, *Vererbung* und *Ereignis* liegen.

Unser Interesse galt dabei insbesondere den folgenden zwei Fragen:

1. Inwieweit kann die Begriffsbildung durch die Arbeit mit Lego-Robotern unterstützt werden?
2. Wie gut eignen sich die von Tim Rinkens entwickelten Programme RCXDownload und RCXDirectMode für den Einsatz in der Schule?

Im Folgenden werden zunächst die Rahmenbedingungen für den Unterricht erläutert, anschließend wird die Durchführung der Unterrichtsreihe dargestellt.

2 Rahmenbedingungen

Der Differenzierungskurs Informatik der Jahrgangsstufe 10 bestand aus 16 Schülern und 4 Schülerinnen. Die SchülerInnen, im Folgenden mit Schüler bezeichnet, wurden seit eineinhalb Jahren von Ruth Dietzel unterrichtet. Die Programmiererfahrungen der Schüler beschränkten sich im Wesentlichen auf die in der Jahrgangsstufe 9 gewonnenen Kenntnisse in der listenorientierten Schulsprache LOGO.

Obwohl die Programmiersprache das Java-Derivat leJOS sein sollte, war es nicht unser vorrangiges Ziel, in die komplexen Strukturen und Möglichkeiten von Java einzuführen. Die Schwierigkeiten, die sich für die Schüler durch den Umstieg auf eine andere Sprache ergeben würden, sollten so weit wie möglich durch vorgegebene Programmteile vermieden werden.

3 Durchführung der Unterrichtsreihe

Im Folgenden werden die einzelnen Sequenzen der Unterrichtsreihe grob skizziert und unsere Erfahrungen mit der Nutzung der Roboter im Informatikunterricht beschrieben.

3.1 Konstruktion der Roboter

Zu Beginn der Unterrichtsreihe stellte die Lehrperson einen Linie-verfolgenden Roboter vor und gab eine kurze Einführung in das Lego Mindstorms Robotics Invention System. Da insgesamt vier Lego-Baukästen in Form einer Leihgabe der Fachschaft Mathematik / Informatik der Universität Paderborn zur Verfügung standen, war es den Schülern möglich, fortan in Fünfergruppen zu arbeiten. Innerhalb der Gruppen mussten die Verantwortlichkeiten für das Zusammenhalten des Materials zugeteilt und ein Gruppenverantwortlicher als Ansprechpartner festgelegt werden. Anschließend bekamen die Gruppen den Auftrag, einen Roboter zu konstruieren, der fahren kann und Tast- und Lichtsensoren besitzt.

Bezüglich der Handhabung der Bauelemente bedurfte es keiner lehrerseitigen Unterstützung, da der Umgang mit Lego-Steinen an die Kindheitserfahrungen der meisten Schüler anknüpfte und sie von der Form der Bauelemente auf deren Funktion schließen konnten. Vermutlich wäre es einträglicher gewesen, kleinere Gruppen zu bilden, doch dies war aufgrund der Anzahl der uns zur Verfügung stehenden Baukästen nicht möglich. So kam es gelegentlich vor, dass sich fünf Konstrukteure in ihrem Baueifer behinderten. Die Aufgabe, einen Roboter zu bauen, der neben Tast- und Lichtsensor über einen möglichst feinmotorigen Antrieb verfügen sollte, stellte sich als schwieriger heraus als erwartet. Daher zogen die Gruppen nach kurzer Zeit die den Kästen beiliegenden Baupläne zu Rate und zeigten sich stets offen für differenzierte Hilfestellungen hinsichtlich verschiedener Konstruktionsmöglichkeiten von Sensoren und Getrieben. Da die Bau-

Anleitungen lediglich Grundgerüste beschreiben, blieb genügend Platz für eigene Kreativität. Unter Umständen wäre es günstiger gewesen, den Schülern von Beginn an einen Plan für einen Grundroboter vorzugeben, um der Gefahr der Überforderung und der Frustration zu entgehen, wenn kein “vernünftiger” Roboter zustande kommt. Die anfängliche Konstruktion durch Versuch und Irrtum beanspruchte derart viel Zeit, dass wir uns nach drei Unterrichtsstunden dazu entschlossen, die Bauphase abzuschließen, indem wir die Roboter im Sinne der Konstrukteure fertig stellten. Mit einer Ausnahme waren alle Gruppen soweit vorangeschritten, dass die Modelle nur hinsichtlich Stabilität und Statik verbessert werden mussten. Letztendlich identifizierte sich jede Gruppe mit “ihrem” Roboter, obwohl sie ihn nicht selbst vollendet hatte.

3.2 Funktionstest der Roboter

Nach der langen Bauphase war es wichtig, die Tauglichkeit der Roboter zu überprüfen, um etwaige Korrekturen vorzunehmen und – im Sinne der Ergebnisorientierung – den Schülern zu ersten Erfolgserlebnissen in Hinsicht auf Steuerbarkeit der Motoren und korrekter Funktionsweise der Sensoren zu verhelfen. An dieser Stelle kam RCXDirect-Mode zum Einsatz. Das Testwerkzeug der RCX-Code-Umgebung der offiziellen Lego-Software wäre sehr viel umständlicher zu bedienen und würde den Gebrauch einer anderen Firmware mit sich ziehen, der wiederum zu Irritationen bei den jugendlichen Anwendern führen kann.

Das Anwenden der Software brachte seitens der Schüler keinerlei nennenswerte Probleme mit sich, lediglich das gleichzeitige Aufladen mehrerer Roboter in einem Raum und der Einfluss der Leuchtstoffröhren beeinträchtigten die Kommunikation über die Infrarot-Schnittstelle. Hier konnte Abhilfe geschaffen werden, indem die Roboter und Transceiver mit den Kartons der Baukästen abgedeckt wurden und man auf das Neonlicht verzichtete.

3.3 Einführung in die Objektorientierte Programmierung

Aufgrund der sehr unterschiedlichen Ausprägungen der Roboter-Modelle sollte jede Gruppe eine Präsentation ihres Roboters vor der Gesamtgruppe vorbereiten. Dazu musste der Roboter auf einem DIN A3 Blatt so genau wie möglich beschrieben werden. Ohne weitere Vorgaben enthielt jede der erstellten Beschreibungen die Eigenschaften und die Funktionalität des jeweiligen Roboters. Eine Gruppe wählte von selbst die Unterteilung in Eigenschaften und Fähigkeiten. Da die vier Roboter ferner Namen von der Fachschaft Mathematik / Informatik bekommen hatten, waren auch diese in den Beschreibungen enthalten.

<p>Alan</p> <p>ein schwenkbares Vorderrad u. zwei Motoren, die die Hinterräder separat voneinander steuern</p> <p>zwei Sensoren: Lichtsensor unter dem Gefährt Tastsensor vor dem Gefährt</p> <p>Fähigkeiten: vorwärts, rückwärts und im Kreis fahren Töne von sich geben</p> <p>eindeutige Merkmale: fünfköpfige Besatzung grüne stylistische Spoiler breites Heck (breite Reifen)</p>

<p>Gottfried Wilhelm</p> <p>4 Räder – vorne klein, hinten groß</p> <p>2 Tastsensoren – vorne</p> <p>1 Lichtsensor – mittig</p> <p>mit Getriebe</p> <p>1 RCX-Baustein</p> <p>linke und rechte Räder jeweils mit einem Motor</p> <p>Fahrtrichtung: vorwärts, rückwärts, links drehen, rechts drehen</p> <p>6 verschiedene Töne</p> <p>alle 3 Sensoren können abgefragt werden</p>
--

Abbildung 1: Beschreibung von zwei Robotern

Die vier Blätter wurden nach der Vorstellung an der Tafel befestigt. Ausgehend von den Robotern als Beispiel für konkrete Objekte und mit Hilfe der Beschreibungen wurde *Objekt* als wichtigster Begriff der Objektorientierung definiert. Um die spätere Arbeit zu vereinfachen, führten wir zusätzlich die Begriffe *Attribut* und *Dienst* ein.

Mit der Begründung, dass nicht alle Eigenschaften für uns interessant waren, wurde eine Schablone für die Beschreibung der Roboter vorgegeben.

Anschließend passte jede Gruppe ihre Beschreibung den Beschreibungskriterien “Was hat der Roboter?”, “Was kann der Roboter?” der Schablone an. Auch diese Version wurde an der Tafel befestigt. So konnte der Begriff der *Klasse* eingeführt werden.

Durch das Anordnen der Objekte als Exemplare eines generischen Roboters hatte sich spontan ein UML-Diagramm an der Tafel ergeben.

<p>Roboter: Alan</p> <p>Anzahl Räder: 3</p> <p>Anzahl Lichtsensoren: 1</p> <p>Anzahl Tastsensoren: 1</p> <p>Anzahl Motoren: 2</p> <p>-----</p> <p>Kann nach vorne fahren</p> <p>Kann rückwärts fahren</p> <p>Kann nach links drehen</p> <p>Kann nach rechts drehen</p> <p>Kann Töne abgeben</p>
--

Abbildung 2:
Ein Roboterobjekt

3.4 Dekonstruktion und Anwendung eines Test-Programms

Gewiss war die Einführung in den Objekt-Begriff anhand realer Objekte vorteilhaft für die Vorstellung der Schüler. Um diesen Vorzug weiter nutzen zu können, gaben wir im Folgenden eine Java-Klasse `Roboter` mitsamt ihrem implementierten Pendant vor. Die Klasse bot lediglich die Dienste `geh_vorwaerts()` und `geh_rueckwaerts()` an. Des weiteren erhielten die Schüler einen Ausdruck der Klasse `Test`, in deren `main`-Methode einige Nachrichten an `robby`, das Roboterobjekt, gesandt wurden. Der Quelltext wurde von den Schülern so weit wie möglich kommentiert, wobei sich herausstellte, dass die `Test`-Klasse im Wesentlichen selbsterklärend war und die Schüler beschreiben konnten, was der Roboter beim Ausführen des Programms machen würde. Im Zuge dieser Dekonstruktion bekamen die Schüler einen ersten Einblick in die Syntax der Sprache und erkannten, wie das Objekt einer Klasse erzeugt und mit der Ausführung von Diensten beauftragt wird.

Die Implementierung sollte nun mittels RCXDownload ausprobiert werden. Die Notwendigkeit des Übersetzens und “Hochladens” war den Schülern durch die frühere Darstellung der Hard- und Software-Schichten des RCX-Bausteins transparent. Sie testeten das Programm und veränderten selbstständig die Nachrichten an das `robby`-Objekt in der `main`-Methode.

Begleitend wurden mehrfach die Begriffe *Objekt* und *Klasse* verallgemeinert, indem zunächst UML-Diagramme zu anderen realweltlichen Bereichen interpretiert und anschließend Klassen und Objekte von den Schülern frei erfunden und in UML-Notation beschrieben wurden. Die Übungen anhand themenübergreifender Gegenstände, wie Fahrrad, Schüleridentität oder Tür diente so der Verinnerlichung des Umgangs mit Klassenhierarchien und sollte auf die bevorstehende ausgeprägtere Hierarchisierung der Roboter-Klassen vorbereiten.

3.5 Nutzung und Realisation von Vererbung

Die Schüler bemängelten recht bald, dass der Roboter nicht links und rechts fahren konnte. Infolgedessen wurde die Klasse `RoboterLR` eingeführt und an einem UML-Diagramm deutlich gemacht, dass diese Klasse sämtliche Attribute und Dienste der Roboter-Klasse enthielt und letztere nur um die Dienste `geh_links()` und `geh_rechts()` erweitert.

Die Definition von Vererbung war für die Schüler zunächst sehr abstrakt, deshalb ließen wir die Lernenden das `robby`-Objekt mit den beiden neuen Diensten beauftragen. Jetzt sollte *Ada*, *Alan*, *Emmy* und *Gottfried-Wilhelm* beigebracht werden, ein Quadrat zu fahren. Als alle Schüler anhand der Fehlermeldungen festgestellt hatten, dass sie die Deklaration und Erzeugung des `robby`-Objektes in der Test-Klasse verändern mussten, um den Roboter auch nach rechts und links fahren zu lassen, und dass dieses Roboterobjekt dennoch vorwärts und rückwärts fahren konnte, obwohl diese Dienste in der Klasse `RoboterLR` nicht implementiert waren, schien förmlich der Groschen in Bezug auf das Konzept der Vererbung gefallen zu sein.

Bei der sich anschließenden Frage, wie wir einen Roboter bekommen können, der den Dienst `geh_quadrat()` anbietet, kam von den Schüler wie selbstverständlich der Hinweis, doch einfach wieder eine Klasse von `RoboterLR` abzuleiten. Die Lehrerin führte kurz ein, wie Vererbung in Java realisiert wird.

Die Schüler waren nun in der Lage, das Prinzip der Vererbung nutzend, eine Klasse `RoboterQuad` zu schreiben, dem `robby`-Objekt die entsprechende Klassenzugehörigkeit zuzuweisen und es mit dem neuen Dienst zu beauftragen. Auf die gleiche Weise wurde eine Klasse entwickelt, die den Dienst zum Abfahren eines Dreiecks bereit stellt.

3.6 Einführung in die Ereignisbehandlung

Die Nutzung des Tastsensors wurde durch die Problemstellung “Roboter soll durch ein Labyrinth fahren” eingeführt. Die Schüler erkannten, dass der Roboter die Fähigkeit besitzen muss, Gegenständen auszuweichen. Durch das Nachvollziehen eines umgangssprachlich formulierten Algorithmus in einem Simulationsspiel bemerkten die Schüler,

dass die Überprüfung des Tastsensors ununterbrochen, auch “während” des Ausführens anderer Befehle, erfolgen muss. Am Beispiel des Tastsensors wurde die *Ereignisquelle* eingeführt, als Ereignisempfänger oder Ereignislauscher ein Objekt der vorgegebenen Klasse *TastZurueckListener*.

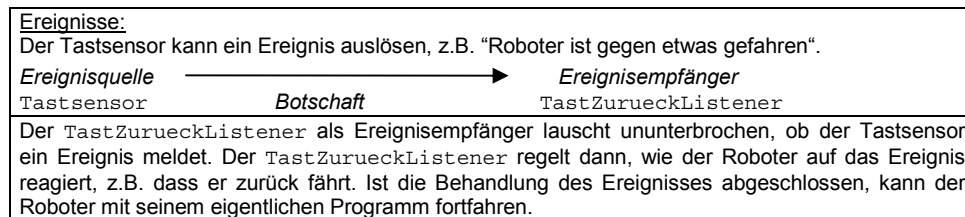


Abbildung 3: Handout an die Lernenden

Der *TastZurueckListener* wurde dem UML-Klassendiagramm hinzugefügt, so dass jedes von der Klasse *RoboterLR* abgeleitete Objekt von ihm profitieren konnte. Anschließend galt es, die Klasse *Test* geeignet anzupassen und zu beobachten, inwieweit sich das Verhalten des *robby*-Objektes verändert hatte.

Leider war es nun an der Zeit, die Roboter zurückzugeben, so dass nur noch eine Stunde blieb, um die Unterrichtsreihe abzuschließen. Wir wollten gewährleisten, dass die Schüler alle Funktionen, die sie den Robotern gegeben hatten, auch einmal ausprobieren konnten. Aus diesem Grunde sollte zum Abschluss der Lichtsensor unter der Problemstellung “Roboter kann eine Linie verfolgen” zum Einsatz kommen.

Angesichts des Zeitmangels und um eine Überforderung der Schüler zu vermeiden, wurde eine von *RoboterLR* abgeleitete Klasse *RoboterLinie* und ein *SpielfeldListener* vorgegeben und nicht weiter auf Hintergründe, wie z.B. die Synchronisation nebenläufiger Prozesse, eingegangen. Durch Transfer der Behandlung des Tastsensors sollten die Schüler hinsichtlich der Aufgabenstellung überlegen, wie die Klasse *RoboterLinie* unter Verwendung eines *SpielfeldListeners* in das Klassendiagramm integriert werden konnte und wie die *Test*-Klasse dementsprechend zu verändern war.

Am Computer hatten die Schüler die Möglichkeit, den Quelltext der Klassen zu betrachten und mit den Klassen zur vorangegangenen Problemstellung zu vergleichen. Der zielbewusste Umgang der Schüler mit den beteiligten Klassen ließ erkennen, dass die Lerngruppe einen guten Überblick über die Objektorientierung und ihre Realisierung bei den Roboterprogrammen in Java bekommen hatte.

4 Reflexion

Vor Beginn der Unterrichtsreihe befürchteten wir angesichts der geringen Vorkenntnisse der Schüler, sie mit dem Umstieg auf eine unbekannte und komplexe Programmiersprache zu überfordern. Diese Bedenken wurden jedoch nicht bestätigt. Der Umgang mit den Robotern förderte die Motivation der Schüler außerordentlich. Die reale Existenz der Roboter-Objekte und die Möglichkeit, sie in die Hand zu nehmen, waren sowohl eine einträgliche Ausgangsbasis für die notwendige Abstraktion in der Objektorientierten

Programmierung als auch ein gutes Anwendungs- und Testmedium für die entwickelten Programme.

Schwierig zu vermitteln war zwar nicht das Ereigniskonzept an sich, aber seine programmiertechnische Umsetzung. Hier gaben wir, auch aufgrund des Zeitdrucks, die implementierten Ereignislauscher vor. Insgesamt deutet der Verlauf der Unterrichtsreihe darauf hin, dass der gewählte Zugang zur Objektorientierung ausbaufähig ist und durchaus die Möglichkeit bietet, weitere Konzepte der Objektorientierung zu vermitteln – vorausgesetzt, es steht genügend Zeit zur Verfügung.

In Bezug auf die von Tim Rinkens entwickelten Schnittstellen konnten wir feststellen, dass die Schüler schnell in der Lage waren, selbständig mit ihnen zu arbeiten und die erstellten bzw. erweiterten Programme an die Robotern zu senden und anschließend zu testen.

Literatur

- [Ba96] Baumann, R.: Didaktik der Informatik. Klett, Stuttgart 1996
- [CH95] Crutzen, C. & Hein, H.-W.: Objektorientiertes Denken als didaktische Basis der Informatik. in: Schubert, S.: Innovative Konzepte für die Ausbildung. Springer-Verlag, Berlin 1995
- [Gu87] Gudjons, H. 1987: Handlungsorientierung als methodisches Prinzip im Unterricht. in: Terhart, E.: Lehr-Lern-Methoden. Juventa, Weinheim 1997
- [HMS99] Hampel, T., Magenheim, J. & Schulte, C. 1999: Dekonstruktion von Informatiksystemen als Unterrichtsmethode. in: Informatik und Schule
- [Hu00] Hubwieser, P.: Didaktik der Informatik. Springer, Berlin 2000
- [KN01] Knudsen, J.; Noga, M.: Das Inoffizielle Handbuch für LEGO MINDSTORMS Roboter. O'REILLY, Köln 2001
- [Le01] The LEGO Group: LEGO Mindstorms. <http://mindstorms.lego.com/> , 2001
- [Ri01] Rinkens, T.: RCXTools for leJOS. <http://rcxtools.sourceforge.net/> , 2001