Institut für Informatik der Humboldt-Universität zu Berlin
Rudower Chaussee 25
12489 - Berlin

Hauptseminar Informatik
Seminarleiter: S. Spolwig
WS 2003/2004

Unterrichtsentwurf / Halbjahresentwurf

Einführung in die Objektorientierte Programmierung

Sebastian Richarz
172234
richarz@informatik.hu-berlin.de

Berlin, 15.04.2004

Inhalt

1. Einleitung

Die vorliegende Arbeit beschreibt einen Lehrplan für das zweite Schulhalbjahr Informatik in der Klassenstufe 11, wobei der Einstieg in die Programmierung rein objektorientiert durchgeführt wird. Der Rahmenlehrplan ist für SchülerInnen der Sekundarstufe II im der 3-jährigen Kursfolge Informatik. In 16 Unterrichtswochen stehen also ungefähr 45 Unterrichtsstunden a 45 Minuten zur Verfügung.

Die Schüler haben in dieser Phase den ersten Kontakt mit der Programmierung. Es gilt daher über der Vermittlung einer Syntax einer Programmiersprache hinaus, grundlegende Vorgehensweisen und Modelle der objektorientierten Programmierung zu behandeln. Wichtig erschien mir auch nicht kleine, so genannte Mickey Maus Programme, für die Vermittlung dieser Inhalte heranzuziehen, sondern den SchülerInnen über die Arbeit an einem komplexeren und realitätsnahen Programm zusätzliche Motivation zu verschaffen. So wird im Verlauf des Schulhalbjahres nach einer Einführungsphase direkt mit der Objektorientierten Modellierung begonnen. Dieses Modelldenken wir durch den Einsatz des Modellierungs-standard Unified modelling language (UML) gefestigt werden. Die Implementation erfolgt in der Programmiersprache Java. Die Wahl dieser Programmiersprache gründet sich auf der universitären Ausbildung von InformatiklehrerInnen. Einen Vergleich Objektorientierter Programmiersprachen hinsichtlich ihrer Eignung für den Informatikunterricht war mir nicht möglich, da ausschließlich Java Inhalt des Informatikstudiums für LehrerInnen ist.

Die Arbeit ist in zwei Teile untergliedert. Der erste Teil beschreibt den Ablauf des geplanten Schulhalbjahres. Er beginnt mit der Begründung, warum aus einer Vielzahl von möglichen Einstiegen die Modellierung als Einstieg in die OOP ausgewählt wurde. Des Weiteren werden Lernvorrausetzungen und angestrebte Lernziele beleuchtet und auf ihre Übereinstimmung mit dem Rahmenlehrplan für Berlin und dessen kommende Änderung überprüft. Im Abschnitt Begründete Stoffauswahl wird dargelegt warum einzelne Themen und Inhalte gewählt wurden und andere wiederum außen vor blieben. Im zweiten Teil der Seminararbeit stellte ich einen Unterrichtsentwurf vor. Dieser Unterrichtsentwurf stellt eine Stunde in der Anfangsphase des Schulhalbjahres mit dem Themenschwerpunkt Modellierung detailliert dar.

2. OOP im Anfangsunterricht

Einleitend möchte ich diesem Abschnitt ein Zitat von LAURA LEMAY[1] voranstellen, welches die Grundintention der Objektorientierten Programmierung im Anfangsunterricht zum Ausdruck bringt.

"Betrachten wir einmal als Vergleich Legos. ... Jedes Bausteinchen hat auf einer Seite kleine Noppen, die in runde Löcher anderer Bausteinchen passen, so dass aus mehreren Legos größere Formen zusammengesteckt werden können. Aus verschiedenen Lego-Teilen (Lego-Rädern Lego-Motoren, Lego-Anhängerkupplungen) können wiederum größere Teile, z. B. Schlösser, Autos oder Riesenroboter, zusammengesetzt werden. Die Kombinationsmöglichkeiten sind schier endlos. Legosteinchen sind kleine Objekte, die auf vorgegebene Weise zusammengesetzt werden können, um größere Objekte zu bilden... Was hat das mit Programmierung zu tun? Alles. Objektorientierte Programmierung funktioniert genau auf diese Weise. Durch Objektorientierte Programmierung besteht ein Gesamtprogramm aus vielen unabhängigen Komponenten (Objekten), die je eine bestimmte Rolle im Programm spielen und die miteinander auf vorgegebene Weise sprechen. "

Weitere Beispiele lassen sich überall in unserer Umwelt finden. So werden Tierarten nach Rassen klassifiziert, wobei jeden Klassen bestimmte Eigenschaften (Attribute) und Fähigkeiten (Methoden) zugeordnet werden. Genauer betrachtet kann man alle Dinge in unserer Umwelt als entsprechendes Objekt betrachten und es mit entsprechenden Eigenschaften und Fähigkeiten klassifizieren. Diese Sichtweise wird von Objektorientierten Programmiersprachen aufgegriffen und in Form von Klassen und Vererbungsbeziehungen informationstechnisch umgesetzt. Die Objektorientierte Sichtweise eignet sich im Besonderen Maße für den Anfangsunterricht, da sie der Denkweise des Menschen sehr nahe ist. Im Vergleich zur Strukturierten Programmierung ist der Entwicklungs-, vor allem Weiterentwicklungsprozess der Programme einfacher, der Wartungsprozess nicht so aufwendig sowie ein schrittweises Annähern des Prototyps an die Erfordernisse möglich. Darüber hinaus liefern Objektorientierte Programmiersprachen in der Regel frühzeitig lauffähige Systeme, was ein besonders wichtiges Kriterium im Anfangsunterricht darstellt. Letztlich ist Produktorientierung und Ernsthaftigkeit der Programme ein Mittel zur Motivation der Schüler und lässt sich erst mit den Vorzügen der Objektorientierten Denkweise im Informatikunterricht unterbringen.

Ein weiterer Vorteil des objektorientierten Entwurfs konnten BAUMFELD und JANNASCH in ihrer Hausarbeit: „Fachdidaktik Informatik“[2] herausarbeiten. Sie stellen fest, dass man ohne ein tieferes Verständnis für die Syntax einer Programmiersprache auskommt. Mehr noch, Analyse und Entwurf legen die Grundlagen einer objektorientierten Programmsyntax und -

[1] *Laura Lemay: Java in 21 Tagen, Markt&Technik, Buch- und Software-Verlag GmbH*

[2] *T.Baumfeld , U.Jannasch: Hausarbeit zum Hauptseminar Fachdidaktik Informatik, HU-Berlin*

semantik, indem beispielsweise grundlegende Begriffe der OOP, wie Objekt, Klasse, Methoden oder Variablen bildhaft und leicht verständlich eingeführt werden oder in UMLed Klassendiagramme in Programmcode überführt werden können. Nach Baumfeld und Jannasch erfolgt das direkte Programmieren des Quellcodes dann im Teilgebiet Objektorientierte Programmierung. Da bis zu diesem Punkt bei den Schülern ein allgemeines Verständnis über den Aufbau von Programmen geschaffen wurde, kann man sich nun primär auf die Probleme der Codeimplementation konzentrieren. Schwerpunktmäßig stehen die Syntax und die Semantik von Java im Vordergrund.

3. Bedingungsanalyse zum Lehrplanentwurf

Die Schulklasse besteht aus 13 Schülern, davon 6 Mädchen und 7 Jungen. Alle SchülerInnen haben das erste Schulhalbjahr besucht. Im Einführungsunterricht wurde eine Einweisung in die Benutzung der vorhandenen Arbeitsmittel (Rechneranlage, Betriebssystem, Vernetzung, Authentifikation, Zugriffsrechte, Verzeichnisbaum, Heimat-Verzeichnis / -Laufwerk). Daneben wurde auch auf aktuelle Anwendungen eingegangen und Einblicke in die geschichtliche Entwicklung der Datenverarbeitung gegeben.
Die Einführung in die Informatik erfolgte über den internetorientierten Zugang nach Rahmenlehrplan[3]. Mit Hilfe dieses Zugangs ...

- wurden theoretische und praktische Grundlagen der Rechnervernetzung und der wichtigsten Internetdienste vermittelt.
- wurden beim Erstellen von HTML-Seiten Objekte, Methoden und Attribute als Begriffe eingeführt.
- wurden die Begriffe Syntax und Semantik von Sprachen am Beispiel HTML erklärt.
- konnten Strategien zur Analyse und Modellierung von komplexen Problemstellungen vorgestellt und geübt werden.
- wird das grundlegende Prinzip der Trennung von Information und Repräsentation eingeführt und an Beispielen erläutert.
- wurden Auswirkungen des Internets auf Wirtschaft und Gesellschaft nicht nur theoretisch, sondern an praktischen Beispielen handelnd erfahren und diskutiert.

Weiterhin wurden Datenbanken und Datenschutz behandelt. Im Vordergrund stand die Benutzung eines einfachen relationalen Datenbanksystems. Neben der programmier-sprachenfreien Benutzung einer Datenbank wurden auch SQL-Scripts zu Abfragen erstellt und dabei mit einfachen Tabellen die Operationen "Selection" und "Join" durchgeführt.

Darüber hinaus haben die SchülerInnen vereinzelt Programmiererfahrungen. Diese sind aber nur unsystematisch vorhanden, da sie im Selbststudium außerhalb der Schule erworben wurden. In Bezug auf Objektorientierte Denkweisen sind Vorkenntnisse aus dem ersten Halbjahr vorhanden. Die Begriffe Objekt, Attribut und Methode sind grob beschrieben worden. Nahezu alle Schüler verfügen über einen heimischen PC mit Internetzugang. Die Klassenräume stellen jedem Schüler einen vernetzten Arbeitsplatz sind zur Verfügung. Die Klassenräume stellen jedem Schüler einen vernetzten Arbeitsplatz sind zur Verfügung. Als Wergzeuge sind UMLed-Designer und Java mit grafischer Oberfläche vorhanden.

[3] *Eckpunkte zum neuen Rahmenplan Informatik SekII für Berlin*

4. Lernziele

Die SchülerInnen sollen nach dem Halbjahr in der Lage sein, ein gegebenes relativ einfaches Problem auf beteiligte Objekte, deren Eigenschaften (Attribute) und Handlungen (Methoden) hin zu analysieren und diesen Sachverhalt in einem UML Klassendiagramm zu modellieren. Daraus soll es den SchülerInnen möglich sein ein Programmgerüst zu erstellen und zum lauffähigen Endprodukt fertig zu programmieren. Um auch schon im Anfangsunterricht komplexere Probleme mit größerer Realitätsnähe behandeln zu können, müssen vom Lehrer vorgefertigte Programmteile oder Programmbausteine zur Verfügung gestellt werden. Dadurch kann sich der Unterricht stärker auf die jeweils neu zu behandelnden Inhalte konzentrieren und auch den für Übungen notwendigen Raum schaffen. Der geschickte Einsatz solcher Programmbausteine ist ebenfalls Anliegen des Unterrichts und bereitet eine effektive Projektarbeit vor. Die Schüler sollen beim Programmieren fertige Programmbausteine und Klassenbibliotheken kennen und einbinden können. Weitere Lernziele sind Stichpunktartig aufgeführt. Die Schüler sollen:

- Begriffe der Objektorientierung verstehen,
- Klassen mit zugehörigen Attributen und Methoden aus realen Problemstellungen bilden können,
- Vererbungsbeziehungen erkennen und in UMLed darstellen,
- Kardinalitäten zuordnen können,
- Syntax der Programmiersprache Java lesen und interpretieren können,
- Den Javacompiler und Interpreter bedienen lernen,
- Anweisungen, Zuweisungen und Operatoren anwenden können,
- Kontrollstrukturen kennen und zur Programmstrukturierung einsetzen können,
- Klassen programmieren und Instanzen durch Konstruktoren aufrufen können,
- Methodenköpfe, Methodeninhalte und die Übergabe von Parametern programmieren lernen,
- Vererbungsbeziehungen in Java programmieren können,
- Klassen von Oberklassen ableiten und Methoden sowie Attribute überschreiben können.

5. Begründete Stoffauswahl

Das in dieser Arbeit beschriebene Halbjahr soll eine Einführung in die Objektorientierte Programmierung bieten. Daher werden nicht alle Teilgebiete der Objektorientierung betrachtet.
Wesentlicher Ansatzpunkt ist hierbei die Objektorientierte Modellierung und Analyse. Die SchülerInnen sollen aus gegebenen Problemstellungen, so genannten Fallstudien, UML-Klassendiagramme erstellen können. Sie sollen also den wichtigen Schritt machen, jedes reale Problem in Objekte mit Attributen, Methoden und Beziehungen untereinander umzuwandeln. Hierbei werden die Fallstudien möglichst so gewählt, dass es nur eine eindeutige Abbildung in der Objektorientierten Sichtweise gibt. Zur Unterstützung wird der UML-Designer UMLed eingesetzt. Bedauerlich ist an dieser Stelle, dass den SchülerInnen kein UML-Tool zur Verfügung steht, das die erstellten Klassendiagramme in Javacode übersetzen kann. Besser geeignet wäre an dieser Stelle zum Beispiel Together© von der Fa. Borland. Die UML Klassendiagramme sollen in Javacode beschrieben werden.
Hieraus und aus der Tatsache, dass Java teilweise umständliche und schwer vermittelbare Sprachkonstrukte braucht um lauffähige Programme zu erzeugen wurden grundsätzlich Programmgerüste, fertige Programmbauteile sowie eine Klassenbibliothek vorgegeben.

PENON und SPOLWIG[4] stellen fest, dass die Unübersichtlichkeit der Klassenbibliotheken von JAVA oft kritisiert wird. Die Strukturierung der Bibliotheken und ihr Umfang ist aber kein sprachinhärentes Problem. Die methodisch-didaktische Planung muss so angelegt werden, dass die Benutzung von Bibliotheken auf ein Minimum bzw. auf spezielle, nach didaktischen Gesichtspunkten entwickelte Pakete reduziert wird.
Ein Vorteil von JAVA ist die Verfügbarkeit: „Da die Basisversion kostenfrei erhältlich ist, kann sie auch an Schüler für Hausarbeiten weitergegeben werden“ [Quelle[4], S. 45]. Zudem ist JAVA auf vielen Plattformen realisiert. Weiter fordert JAVA vom Programmierer überwiegend die explizite Angabe von implizit getroffenen Entscheidungen. Während man dem Anfänger umfangreiche Klassenbibliotheken noch vorenthalten sollte, bieten sie dem Fortgeschrittenen die Möglichkeit auch bei weiterführenden Themenstellungen innerhalb der Informatik JAVA als Werkzeug zu benutzen. Neben diesen Fakten ist JAVA eine Modesprache bei Computerinteressierten, gerade durch die Verbreitung im Internet. So ist ein starkes Interesse bei Informatikschülern zu erwarten: „Die Programme lassen sich leicht ins WWW einbinden, was erheblich zur Motivation der Schülerinnen und Schüler beiträgt“ [Quelle[4], S. 45]. Dabei ermöglicht JAVA die Einbindung von multimedialen Inhalten mit relativ geringen Mitteln, was zu einer weiteren Akzeptanz bei der Zielgruppe führt.

[4] Penon, Johann; Spolwig, Siegfried: Schöne visuelle Welt?
Objektorientierte Programmierung mit DELPHI und JAVA.
In: LOG IN 18 (1998) Heft 5

Nach dem Ansatz: „Stifte und Mäuse“ können die Schüler einzelne Klassen mit deren Methoden und Attributen nutzen. Von besonderer Bedeutung ist hierbei, dass die grafische Darstellung nicht über eine Klasse (z.B. Bildschirm bei Stifte und Mäuse) erfolgt, sondern sich jede Klasse selber darstellen kann. Hierzu müssen von der Lehrkraft geeignete Methoden für die jeweiligen Klassen bereitgestellt werden.
Nach Kenntnis von Klassendiagrammen und deren Erzeugung wird das Softwareprojekt Billard vorgestellt. Zunächst eine Billardkugel soll sich auf einem definieren Billardtisch bewegen und von den Wänden zurückgeworfen werden. Hierzu müssen die Klassen Billardtisch und Kugel erstellt werden. Beide Klassen müssen eine Methode „darstellen“ haben. Diese Darstellungsmethode muss von der Lehrkraft maßgeblich vorgegeben werden indem ein JavaApplet erstellt wird, dass Billardtisch und Kugel anzeigen kann.
Syntax von Java soll beim Umsetzen des Softwareprojekts „nebenbei“ erlernt werden. Dass dies im Verlauf des Projektes nicht immer möglich ist und auch erheblich vom Leistungsstand und Leistungsbereitschaft der Schülerschaft abhängt erfordert von der Lehrkraft jederzeit fachlich versierte Vorträge einschieben zu können. Den Schülerinnen müssen hinreichend Informationsquellen zur Verfügung stehen um auftauchende Fragen selber klären zu können. Die Implementierungsphase ist also geprägt durch selbstständiges Arbeiten am Computer. Die Lehrkraft steht beratend zur Seite. Diese Arbeitsform schult die Team- und Kommunikationsfähigkeit der SchülerInnen.

6. Halbjahresablauf

Tabelle 1: Halbjahresablauf Einführung in OOP

Thema	Inhalte	Anmerkungen	Zeit in h
Einführung	• Vorstellung • Ablauf des Schulhalbjahres		1
Einführung, Grundlagen der Objektorientierten Modellbildung	• Einführung in Modellbegriff • Motivation: Warum modellieren • Widerholung des Aufbau eines Computers	Aus den Komponenten eines Computers soll ein Modell entwickelt werden. Bildschirm, Tastatur, Maus werden als Objekte vorgestellt.	2
	• Reale Objekte werden beschrieben • Vorstellung der Begriffe Attribute und Methoden		3
Objektorientierte Analyse und Entwurf	• Vorstellung des UML Editors UMLed • Erste Übungen	Vorgegebene Objekte mit zugehörigen Methoden und Attributen werden abgebildet	2
	• Vererbung, Klassen, Objekte (Instanzen), Datentypen, Variablen • Kardinalitäten (Aggregationen, Assoziationen, Generalisierung, Komposition)	Übung an Beispielen *Hier ist u.a. die Durchführung der geplanten Unterrichtsstunde angesiedelt (Abschnitt 7)*	4
Softwareprojekt Billard	• Problemstellung erläutern	Eine Billardkugel soll sich auf einem definieren Billardtisch bewegen und von den Wänden zurückgeworfen werden.	1
	• Modellbildung, Umsetzung in UMLed	In dieser Phase sollen die SchülerInnen ein vollständiges Modell von Billard erstellen; genauer Klassen mit Attributen und Methoden definieren und die Beziehungen untereinander klären	6
Implementierung Billard	• Einführung in Java	Im Lehrervortrag werden Grundlagen der Syntax von Java sowie die zur Verfügung gestellten Programmbausteine und Klassenbibliotheken erläutert	3
	• Einführung in BlueJ	Applets erstellen, Neues Projekt anlegen, Datentypen festlegen, Compiler und Interpreter	3

Thema	Inhalte	Anmerkungen	Zeit in h
Im folgenden werden Schrittweise Grundlegende Strukturen der Programmiersprache vermittelt			
	• Zuweisungen (Vergleiche, Operatoren, Arithmetik, Boolsche Operatoren) • Bedingte Anweisungen (if – then – else, switch) • Schleifen (while, for)		6
	• Klassen und Instanzen	Attribute, Methoden, Konstruktoren und Destruktoren, Parameterübergabe	4
	• Gültigkeitsbereich von Bezeichnern	Public, Private	3
	• Vererbungs-Beziehungen	Hierbei soll Polymorphismus nur in Grundzügen und Überladen von Methoden außen vor gelassen werden	2
	• Vervollständigung des Softwareprojektes Billard	Endergebnis soll ein lauffähiges Java-Applet sein. Sollten der vorangegangene Stoff schneller abgehandelt werden sind in dieser Phase Erweiterungen möglich (z. B. mehrere Billardkugeln die miteinander zusammenstoßen können)	4
Leistungskontrolle			2
		Summe:	46

7. Unterrichtsentwurf

7.1 Stellung der Stunde im Unterricht und Schülervoraussetzung

Die Schulklasse besteht aus 13 Schülern, davon 6 Mädchen und 7 Jungen. Die SchülerInnen zu motivieren ist schwierig, da die allgemeine Disziplin schlecht ist und es ihnen schwer fällt sich zu konzentrieren. Peter spielt im Klassenverband eine besondere Rolle. Er gilt bei den SchülerInnen als „Computerfreak“, da er häufig in der Firma seines Vaters bei der Erstellung professioneller Firmenhomepages mitarbeitet. Er ist desinteressiert und äußert den Sinn dieser Einführung in die Objektorientierte Programmierung in Bezug auf Ernsthaftigkeit und Realitätsnähe an.
Alle SchülerInnen haben das erste Schulhalbjahr besucht. Im Einführungsunterricht wurde eine Einweisung in die Benutzung der vorhandenen Arbeitsmittel (Rechneranlage, Betriebssystem, Vernetzung, Authentifikation, Zugriffsrechte, Verzeichnisbaum, Heimat-Verzeichnis / -Laufwerk). Daneben wurde auch auf aktuelle Anwendungen eingegangen und Einblicke in die geschichtliche Entwicklung der Datenverarbeitung gegeben. Tiefere Einblicke in die bereits behandelten Themen gebe ich im Abschnitt 3 – Bedingungsanalyse zum Lehrplanentwurf.
Im laufenden Schulhalbjahr haben die SchülerInnen eine Einführung in die Grundlagen der Objektorientierten Modellierung erhalten und gemeinsam mit der Lehrkraft erste Übungen zur Modellierung gemacht. Die Begriffe Klassen, Instanzen (Objekte), Attribute, Methoden und Vererbung würden behandelt und deren Funktion in der Objektorientierung erläutert. Sie können mit dem UML-Editor UMLed umgehen und haben das Modellieren mit UMLed geübt. Die geplante Unterrichtsstunde soll den Umgang mit Modellen festigen. Eine komplexe Aufgabenstellung bzw. Fallstudie soll mit UMLed in ein geeignetes Objektorientiertes Modell umgesetzt werden. Ergebnis soll ein Klassendiagramm in UML sein.

7.2 Sachanalyse

Eine ***Klasse*** ist allgemein eine Gruppe von Dingen, Lebewesen oder Begriffen, die gemeinsame Merkmale oder Eigenschaften haben. Die eigentlichen Objekte (Instanzen) lassen sich aus einer Klasse erzeugen. In UMLed werden Klassen als dreigeteilte Rechtecke dargestellt. Die Felder beschreiben Klassenname, Attribute und Methoden der Klasse (Bild 1).

Bild 1: UMLed-Klassendarstellung

Die ***Vererbung*** ist eines der wichtigsten Prinzipien der objektorientierten Programmierung. Man versteht darunter, dass von bereits bestehenden Klassen neue abgeleitet werden können, die zunächst die gleichen Eigenschaften und Methoden besitzen wie ihre Vorgänger. Zusätzlich werden sie jedoch noch mit neuen Elementen ausgestattet, die entweder Erweiterungen ihrer „Eltern“ darstellen oder aber alte Elemente überschreiben. Diejenige Klasse, von der eine neue Unterklasse erzeugt wird, nennt man Oberklasse. Eine Mehrfachvererbung wie bei C++, bei der eine Unterklasse Elemente von mehreren Oberklassen erbt, gibt es in Java nicht. Die Mehrfachvererbung kann nur mit Hilfe von Interfaces realisiert werden und soll im Rahmen dieser Unterrichtplanung weggelassen werden. Von der Klasse **Fahrzeuge** könnten neue Unterklasse abgeleitet werden, die zusätzliche Methoden und Attribute besitzt. Dadurch wird es auch möglich, ganz neue Klassen von Autos abzubilden, wie zum Beispiel **LKWs**, **PKWs** und **Motorräder**. Durch Vererbung wird die Forderung nach leicht erweiterbaren Programmkomponenten für die Softwareentwicklung erfüllt.

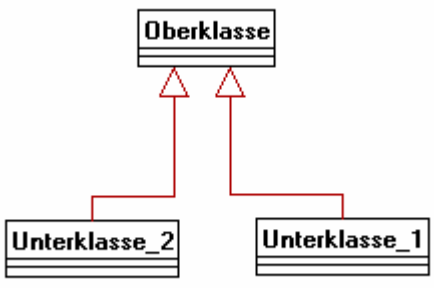


Bild 2: IST-Beziehung in UMLed

Vererbungsbeziehungen werden in UMLed IST-Beziehung genannt. Eine IST-Beziehung ist eine Beziehung zwischen einer allgemeinen Klasse (der Oberklasse) und einer speziellen Klasse (der Unterklasse). Die Unterklasse erbt die Attribute und Methoden der Oberklasse, besitzt aber häufig zu denen der Oberklasse weitere Attribute und Methoden. Geerbte Attribute und Methoden werden in der Unterklasse nicht noch einmal aufgeführt. Besitzt aber eine Methode der Unterklasse dieselbe Signatur (dasselbe Aussehen) wie eine Methode der Oberklasse, wird die Methode der Oberklasse überschrieben, man spricht in diesem Fall von Polymorphie. Grafisch wird die IST-Beziehung über durchgezogene rote Pfeile, die mit einer Spitze auf die Oberklasse weisen, dargestellt (Bild 2).

7.3 Begründete Stoffauswahl

Den SchülerInnen wird eine komplexe Problemstellung vorgegeben. Das Thema: „Problemorientierte Verkehrsmodellierung auf Bundesautobahnen“ ist einem aktuellen Forschungsvorhaben der Firma DaimlerChrysler AG entnommen und soll so motivierend auf die SchülerInnen wirken. Da das Objektorientierte Modellieren ein wesentlicher Bestandteil im Lehrplan des Schulhalbjahres ist, soll es mit dieser Übung vertieft werden. Besonders das Prinzip der Vererbung soll verdeutlicht werden.

7.4 Lernziele

Die Schüler sollen:

- Durch die Realitätsnähe und Ernsthaftigkeit des gewählten Fallbeispiels motiviert werden,
- die einzelnen Objekte des Fallbeispiels klassifizieren,
- Vererbungsbeziehungen zwischen den einzelnen Klassen erkennen,
- Klassen und Vererbungsbeziehungen mit UMLed darstellen/umsetzen,
- Den Klassen Attribute und Methoden zuordnen,
- Erkennen welche Vorteile das Prinzip Vererbung hinsichtlich Übersichtlichkeit und Erweiterbarkeit mit sich bringt,
- Den Umgang mit UMLed festigen.

7.5 Weg- und Medienentscheidung

Zu Beginn dieser Unterrichtseinheit werden die SchülerInnen in die Rolle von Mitarbeitern einer großen Firma versetzt. Jede/r SchülerIn erhält einen Auftragszettel mit der Problemstellung. Nach einer anfänglichen Fragerunde sollen die SchülerInnen ihren Arbeitsauftrag weitestgehend selbstständig auf Papier ausarbeiten. Die Lehrkraft steht beratend zur Seite.
In einer zweiten Phase werden die Entwürfe diskutiert und an der Tafel zu einem gemeinsamen Modell zusammengetragen. In der Schlussphase wird das Objektorientierte Modell im UMLed umgesetzt. Hierzu ist Gruppenarbeit mit bis zu 3 Schülern pro Gruppe angedacht. Abschließend erfolgt in der nächsten Unterrichtstunde eine Betrachtung und Schlussdiskussion der Ergebnisse. Durch die zeitliche Trennung von Arbeit am UML-Modell und Auswertung werden die SchülerInnen gezwungen sich noch mal, und damit tiefer, mit dem Thema zu befassen. Neu auftretende Fragen können geklärt werden.

7.6 Tabellarischer Unterrichtsverlauf

Dauer [min]	**Inhalt**	**Medien**	**Lehrform**	**Sozialform**
2	• Einleitung; • Schüler werden in die Rolle von Firmenmitarbeitern versetzt	OHP	Lehrervortrag	Frontal-unterricht
5	• Anfängliche Fragen werden geklärt	Tafel, OHP	Gespräch	
20	• SchülerInnen entwickeln ein OO Modell aus dem gegebenen Problem	Papier und Stift	Schülerarbeit, Lehrer ist Berater	Einzelarbeit der Schüler
5	• Die erarbeiteten Modelle werden zu einem gemeinsamen Modell zusammengetragen	Tafel	Gespräch, Ideensammlung	Lehrer-Schüler-interaktion
13	• Umsetzung in UMLed	Computer	Schülerarbeit, Lehrer ist Berater	Gruppen-arbeit

7.7 Anhang

7.7.1 SchülerInnenauftrag

DAIMLERCHRYSLER

Auftrag

Sie sind in Mitarbeiter bei der Firma DaimlerChrysler in der Projektgruppe: „Problemorientierte Verkehrsmodellierung auf Bundesautobahnen“

Im Projekt wird die technische Realisierung von fahrzeug-basierten ad hoc Kommunikationsnetzen im Straßenverkehr untersucht. Da die Datenübertragungs-reichweiten der Fahrzeuge beschränkt sind, spielen die Abstände zwischen den Fahrzeugen eine große Rolle. Die relativen Geschwindigkeiten der Fahrzeuge beschreiben, wie schnell sich diese Abstände ändern.

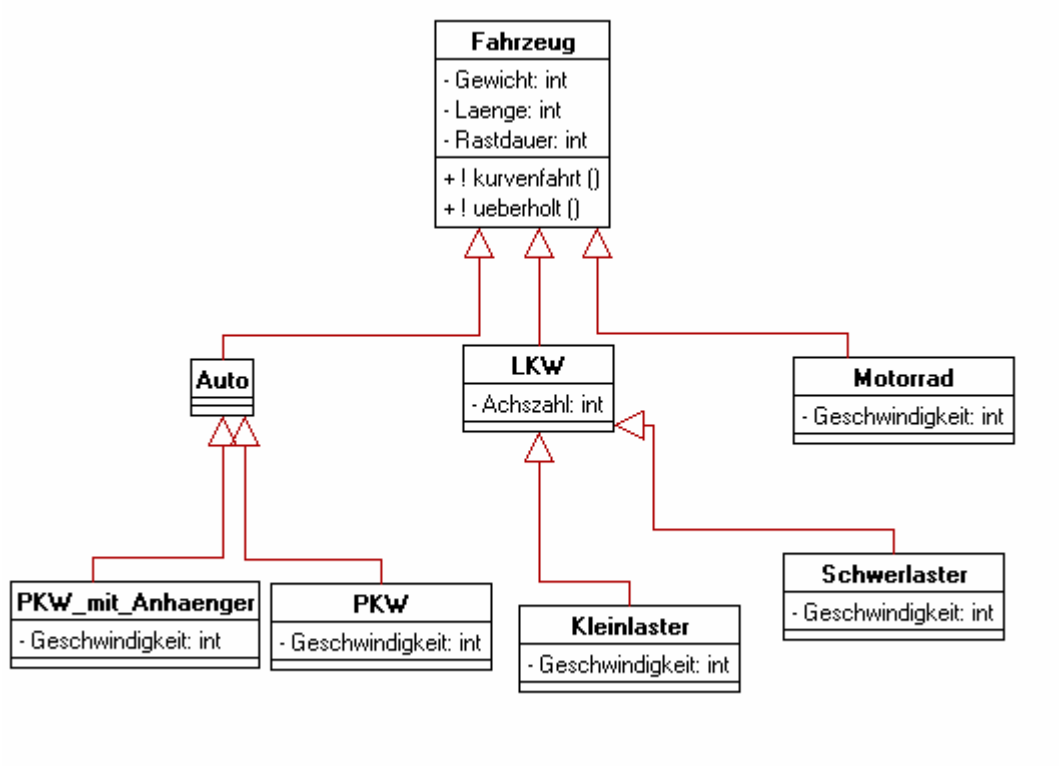
Basierend auf fundamentalen Beziehungen im Straßenverkehr und Verkehrsmessdaten sollen die Bausteine einer problemorientierten einfachen Verkehrsmodellierung zur Erzeugung von realistischen Abständen und Geschwindigkeiten auf Bundesautobahnen in einem Simulationsprogramm abgebildet werden. Sie sind noch am Anfang des Projektes und benötigen ein Objektorientiertes Modell der Fahrzeuge auf Bundesautobahnen. Bilden sie die möglichen Fahrzeuge auf einer Autobahn möglichst genau nach. Vergeben Sie alle relevanten Attribute und Methoden zur Späteren Umsetzung des geforderten Simulationsprogramms.

Ein wichtiger Punkt um die Simulation realistisch zu gestalten ist das Fahrerverhalten. Vom statistischen Bundesamt haben sie Informationen über:

- Durchschnittsgeschwindigkeiten verschiedener Fahrzeugtypen.
- Geschwindigkeitsverhalten auf kurvenreichen Strecken
- Geschwindigkeitsverhalten aller Verkehrsteilnehmer bei Nacht in Vergleich zum Tag
- Durchschnittliche Verweildauer auf Autobahnen (hieraus resultieren andere Geschwindigkeiten wegen Autobahnauf- und abfahrten)
- Verweildauer auf Rastplätzen

Versuchen Sie diese Informationen bei der Modellierung zu berücksichtigen.

7.7.2 Lösung: UMLed Darstellung



8. Literatur

A. Niemann	Das Einsteigerseminar Objektorientierte Programmierung in Java, Verlag Moderne Industrie Buch AG & CO, Landsberg, 2001
L. Lemay:	Java in 21 Tagen, Markt &Technik, Buch- und Software-Verlag GmbH
T.Baumfeld, U.Jannasch:	Hausarbeit zum Hauptseminar Fachdidaktik Informatik, HU-Berlin
	Eckpunkte zum neuen Rahmenplan Informatik SekII für Berlin
J. Penon, S. Spolwig:	Schöne visuelle Welt? Objektorientierte Programmierung mit DELPHI und JAVA. In: LOG IN 18 (1998) Heft 5
A. Schwill:	Programmierstile im Anfangsunterricht, Facharbeit Uni Paderborn
T. Cremer:	Einführung einer objektorientierten Programmiersprache in der Sekundarstufe II, Examensarbeit im Fach Informatik für das Lehramt (Sekundarstufe II), Uni Münster, http://www.muenster.de/~cremer/mitJava/javaLernen/Examen1/EinfuehrungOOP.PDF
D. Vollmer, A. Hiller:	Problemorientierte Verkehrsmodellierung auf Bundesautobahnen im Rahmen des Projektes F L E E T N E T, Mai01, DaimlerChrysler AG, http://www.csv.ica.uni-stuttgart.de/homes/ph/jobs/vollmer/Verkehrsmodellierung.pdf