

Rafael Schreiber

2. Schulpraktisches Seminar Marzahn-Hellersdorf

Berlin, den 3.11.2004

Entwurf einer Unterrichtsstunde zur Zweiten Staatsprüfung für das Amt des Studienrats

Fach: Informatik

Zeit: 3.11.2004 , 10.00 – 10.45 Uhr

Ort: Leonard- Bernstein-Gymnasium

Lerngruppe: WPU 10

Raum: 2112

Prüfungsvorsitzende: Herr Befort

Hauptseminarleiter: Frau Sauerbaum-Thieme

Schulleiter: Herr Dr. Militzer

Fachseminarleiter Deutsch: Herr Dr. Grenzler

Fachseminarleiter Informatik: Herr Witten

Lehrervertreter: Herr Paetzold

Aufgabe: Planen und erteilen Sie eine Unterrichtsstunde aus der Reihe „*Einführung in die Objektorientierung mit Java*“. Bestimmen Sie den Schwerpunkt selbst.

Gliederung des Unterrichtsentwurfs

1. Thema und Einbettung in die Unterrichtsreihe
2. Unterrichtsvoraussetzungen
3. Sachstrukturanalyse
4. Didaktische Reduktion
5. Lernziele
6. Didaktisch-methodische Erwägungen
7. Verlaufsplanung
8. Literatur
9. Anhang (Arbeitsblätter, Folien, Diagnostische Matrix der Lerngruppe)

1. Thema

1.1 Thema der Unterrichtsreihe

Einführung in die Objektorientierte Programmierung mit Java

1.2 Thema der Unterrichtsstunde

Vererbung als grundlegendes Konzept der Objektorientierung

1.3 Einbettung der Unterrichtsstunde in die Unterrichtssequenz

1.	Der Algorithmusbegriff in der Informatik
2.	Java und seine Historie
3.	Klassen und Objekte
4.	Einführung in die Entwicklungsumgebung BlueJ
5.	Aufbau von Klassen in Java
6.	Zustand von Objekten: Attribute
7.	Datentypen in Java
8.	Zustandsveränderung von Objekten: Methoden
9.	Methoden mit und ohne Rückgabotyp
10.	Erstellung und Verwendung von Methoden (Übungsstunde)
11.	Parameter in Methoden
12.	Objektinteraktion
13.	Initialisierung von Objekten: Konstruktor
14.-15.	Programmierte Erstellung von Objekten
15.-16.	Erstellen von Klassen. Komplexe Anwendung von Klassen und Objekten
17.-18.	Zweistündige zentrale Klassenarbeit der WPU-Kurse
19.	Auswertung Klassenarbeit
20.	Medium - Planung einer Medienbibliothek
21.	Vererbung als grundlegendes Konzept der Objektorientierung
22.	Anwendung und Übung von Vererbung
23.-24.	Polymorphismus
25.-26.	Anwendung im Internet: Java-Applets

2. Unterrichtsvoraussetzungen

2.1. Allgemeine Unterrichtsvoraussetzungen

Die Lerngruppe ist ein klassenübergreifender Wahlpflichtkurs der 10. Jahrgangsstufe, der seit Beginn des Schuljahres selbstständig von mir unterrichtet wird. In diesem Kurs lernen elf Schüler, alles Jungen. Die Schüler lernten in dieser Zusammensetzung bereits in der neunten Jahrgangsstufe zusammen. Inhaltlicher Schwerpunkt des Unterrichts in der neunten Klasse war Einführung in die Algorithmik mit Hilfe von Turbo Pascal.

In dieser Lerngruppe zeigen sich deutliche Leistungsunterschiede. Mindestens drei Schüler, Steffen, Ben und Felix, programmieren in ihrer Freizeit und beschäftigen sich auch sonst viel mit dem Rechner (z.B. Erstellen von 3D-Modellen, Web-Programming/PHP/SQL). Ben und Felix scheinen viele der zu behandelnden Gegenstände in der momentanen Unterrichtsreihe zur Objektorientierung teilweise bis sehr vertraut zu sein.

Trotz des hohen Anspruchs und der Schwierigkeiten, die oft beim Umstieg von der prozeduralen auf die objektorientierte Sichtweise entstehen, sind alle Schüler motiviert und interessiert. Inzwischen entstandene Wissenslücken führen jedoch insbesondere bei Franz und Jabin zu gelegentlich demotivierender Überforderung, die aber streckenweise durch geeignete Sozialformen aufgefangen werden kann.

2.2 Spezielle Unterrichtsvoraussetzungen

Diese Stunde ist die 21. der Reihe, welche durch die Relevanz der Thematik einen überdurchschnittlichen Reihenumfang hat. Die Schüler kennen inzwischen die Struktur und Funktionsweise von Klassen und Objekten und können selbstständig Klassen und Klasselemente erstellen und verwenden. Trotz steter Übung haben manche Schüler gerade bei Methoden und Parametern noch immer Schwierigkeiten in der Unterscheidung von Deklaration und Verwendung. Die Schüler kennen Zugriffsmodifikatoren und verwenden diese. Unklar war in diesem Zusammenhang bisher die Funktion von *protected*. Als Java-Lernumgebung wird *BlueJ* verwendet, welches eine grafische Darstellung von Java-Klassen und -objekten ermöglicht und speziell für die Lehre entwickelt wurde.

Die Schüler sind mit kooperativen Sozialformen vertraut. Gruppenarbeit ist eine in diesem Kurs von mir häufig eingesetzte Form zur Erarbeitung und Sicherung. Partnerarbeit wird zusätzlich durch die begrenzte Rechneranzahl initiiert. Es sind 9 Rechner im Informatikraum, der wegen Umbauten des Hauses 1 ein anders genutzter Deutschraum ist. Ein lokales Netzwerk ist vorhanden, Internetzugriff ist jedoch nicht möglich, was sich für selbstständige Recherche als sehr hinderlich erwiesen hat.

Thematisch wurde im Hinblick auf Vererbung und Polymorphismus die Planung der Modellierung einer Medienbibliothek angeregt, die mit einer Klasse *Medium* und Diskussion über den *Mediums*begriff im gestrigen Unterricht begann. Die dort erstellte Klasse *Medium* wird (in reduzierter Form) als Basis und Ausgangspunkt für die heutige Stunde dienen.

3. Sachstrukturanalyse

Für die formale Modellierung von informatischen Problemen hat sich in den letzten Jahrzehnten das objektorientierte Paradigma etabliert. Wesentlicher Kern desselben ist die Aufhebung der Trennung von Daten und den sie verarbeitenden Algorithmen. Zusammengehörige Daten und Algorithmen werden in Klassen zusammengefasst und damit von der Umgebung abgekapselt. Diese Klassen wiederum sind abstrakte Beschreibungen konkreter Ausprägungen ihrer selbst, **der Objekte**, welche Eigenschaften (die Daten) und Verhaltensweisen (die Algorithmen) zusammen enthalten. Neben der genannten Kapselung in Klassen ist die **Vererbung** ein Basiskonzept der Objektorientierung. Auf Modellierungsebene ermöglicht sie die Herstellung von hierarchischen Relationen zwischen Klassen.

Vererbung bedeutet die Möglichkeit der Übernahme von Eigenschaften (sog. Attributen) und Verhaltensweisen (Methoden) einer Klasse aus einer anderen bereits existierenden Klasse. Das heißt, alles, was eine Klasse hat und kann, hat und kann dann auch die erbende Klasse. Durch die Möglichkeit, von erbenden Klassen wiederum zu erben, lassen sich komplexe hierarchische Strukturen aufbauen. Nicht möglich ist es für eine erbende Klasse (Subklasse) in Java jedoch, von mehreren Elternklassen (Superklassen) zu erben. Diese **Mehrfachvererbung**, welche man z.B. in C++ findet und die eine nahezu unendlich komplexe Strukturabbildung ermöglicht, wurde aufgrund der im gleichen Maß schwieriger werdenden Verständlichkeit sowie kaum nachzuvollziehender Seiteneffekte in Java nicht implementiert.

Die entscheidenden **Vorteile von Vererbung** basieren alle auf der damit zu erreichenden besseren Strukturierung auf einem höheren Abstraktionsniveau, einer „saubereren“ Abbildung der Welt. Die Codemenge wird bei gleichzeitig vereinfachter Wartung und Erweiterbarkeit verringert, womit sich Fehler verringern und sich der Nutzwert des Programms erhöht.

Die Vorteile der Vererbung sind im Einzelnen:

- **Die Vermeidung von Code-Duplizierung**
Die Verwendung von Vererbung vermeidet, dass gleiche oder ähnliche Codeabschnitte mehrfach geschrieben werden müssen.
- **Einfachere Wartung**
Eine Veränderung der bestehenden Attribute und Methoden, die von mehreren Klassen geteilt werden, muss nur noch einmal vorgenommen werden. Dies ist gerade bei Wartung durch einen anderen als den ursprünglichen Programmierer/Modellierer sinnvoll.
- **Erweiterbarkeit**
Das Hinzufügen von Elementen zu Klassen bzw. Klassenstrukturen wird durch die Wiederverwendung von Quelltext erheblich vereinfacht.

Vererbungsrelationen kann man immer in Form von „**ist-ein**“- **Relationen** ausdrücken, so z.B.

- *Ein Pferd ist ein Tier.*
- *Ein Buch ist ein Medium.*
- *Ein Reihenhaus ist ein Haus ist ein Gebäude ist ein von-Menschen-erbautes-Betretbares.*

Sie sind zu unterscheiden von anderen Beziehungen, die zwischen Objekten der Welt (wie auch ihren Abbildungen im Rechner) herrschen, z.B.

- der „ist-Teil-von“-Beziehung (*Lenkrad ist ein Teil von einem Auto.*)
- der „hat“-Beziehung (*Ulli hat eine Schwester.*)
- oder der „enthält“-Beziehung (*Ein Bücherregal enthält Bücher.*)

Zu beachten ist, dass ein Satz wie „Herr Schreiber ist ein Mensch“ keine gültige Vererbungsrelation ausdrückt, da „Herr Schreiber“ ein konkretes Objekt darstellt, die informatische Vererbung sich jedoch auf Klassen, also Kategorien von Dingen bezieht.

Subklassen können die von den Superklassen geerbten Elemente erweitern, sie sind nicht auf die geerbten Elemente beschränkt. Diese Sichtweise der Vererbung von der Subklasse aus findet sich auch bei der konkreten Klassendeklaration in dem Schlüsselwort *extends* (erweitert) wieder:

```
//Beispiel Superklasse Tier
public class Tier{
    private int höhe_in_cm;
    private String farbe;
}

//Beispiel Subklasse Pferd
public class Pferd extends Tier {
    private String rasse;
    // ...weitereKlassenbeschreibung}
```

Ein erzeugtes Objekt der Klasse *Pferd* hat nun sowohl die Eigenschaften *größe_in_cm* und *farbe* von *Tier* als auch das im Beispiel zusätzlich definierten Attribute *rasse*.

In gleicher Weise werden **Methoden** vererbt. Diese stehen dann der Subklasse zur Verfügung. Die Subklasse kann aber, wenn sie will, die geerbte Methode mit einer eigenen gleichen Namens überschreiben. Dieses *Overriding* ist ein Aspekt von Polymorphismus, der möglichen Vielgestalt einer Klasse.

Der dritte Bestandteil einer Klassendefinition, der **Konstruktor**, wird nicht vererbt. Konstruktoren werden bei der Erstellung von Objekten aufgerufen und dienen deren Initialisierung. Die erbende Klasse muss einen expliziten Konstruktoraufruf ihrer Superklasse durchführen, damit ihr die ererbten Attribute und Methoden zur Verfügung stehen. Dieser erfolgt mit dem Befehl **super()** sowie eventuellen Parametern und befindet sich im eigenen Konstruktor. Wird der *super()*-Aufruf nicht getätigt, ruft Java automatisch den Standardkonstruktor auf, was jedoch nur solange zum Erfolg führt, wie die Superklasse ohne Parameter deklariert ist.

Auch von Superklassen können konkrete Objekte erstellt werden. Soll dies nicht geschehen, sondern dienen diese nur zur Strukturierung und Vererbung an Subklassen, nennt man sie abstrakte Klassen. Durch das Schlüsselwort *abstract* kann ihnen explizit die Objektableitung verboten werden.

4. Didaktische Reduktion

Das Zentrale der Vererbung ist die Übernahme von bereits existenten Klassenelementen bei der Erstellung neuer Klassen. Diese Wiederverwendung spart Code, verringert Arbeit und Wartungsaufwand. Vererbung erhöht die Effektivität von Programm- und Datenstrukturen. Für die Stunde habe ich mich für ein Vorgehen mithilfe von Programmcode und dessen Analyse entschieden, da dies eine spätere Umsetzung am Rechner und damit eine für die Schüler fassbare Demonstration eines abs-

trakten Prinzips ermöglicht. Alle Spezialfälle und Seitenbetrachtungen der Vererbung werden didaktisch reduziert. Das betrifft:

- Vererbung von Konstruktoren (Sie sind auch technisch wegen des Standard-konstruktoraufrufs *super()* reduzierbar.) Ihre Reduktion in einer Einführungsstunde ist wegen des Ausnahmeverhaltens von Konstruktoren absolut notwendig.
- Zugriffsmofifikatoren (Sie sind auch technisch wegen der standardmäßigen public-Behandlung reduzierbar.) Ihre Reduktion in einer Einführungsstunde ist sinnvoll wegen der bei Vererbung auftretenden Zugriffsproblematik.
- Mehrfachvererbung (ist ein Spezialfall, der nicht behandelt werden muss.)

Es erfolgt eine Konzentration auf die Vererbung von Attributen. Nicht verzichtet werden soll auf das Weitervererben von ererbten Attributen an eine noch spezifischer Klassenebene, da dies eine hervorragende Progression zu ermöglichen scheint.

5. Lernziele

Grobziel: Die Schüler kennen das Konzept der Vererbung.

Feinziele:

- FZ1: Die Schüler erkennen, dass eine Klasse Medium nicht alle Eigenschaften ihres Spezialfalls CD-Rom haben kann, indem sie das Hinzufügen eines weiteren Attributs diskutieren.
- FZ2: Die Schüler kennen die Syntax und Semantik von erbenden Klassen, indem sie einen Quellcode nachvollziehen und die dortigen Attribute als übernommen interpretieren.
- FZ3: Die Schüler wissen, dass die Attribute der Oberklasse auch zur Unterklasse gehören, indem sie sagen, dass die Methode *ausgeben()* Eigenschaften von CDROM ausgibt.
- FZ4: Die Schüler können das Verhältnis von Unterklasse und Oberklasse beschreiben, indem sie für das Wort *extends* einen adäquaten deutschen Ersatz finden.
- FZ5: Die Schüler können ein Vererbungsverhältnis abstrakt erkennen und beschreiben, indem sie eine schematische Darstellung interpretieren und erklären.
- FZ6: Die Schüler wenden das Vererbungsprinzip an, indem Sie am Rechner eine Subklasse erstellen, diese auf die richtige Superklasse beziehen und alle vorhandenen Attribute ansprechen. (fakultativ)

6. Didaktisch-methodische Überlegungen

Das Vererbungskonzept soll sich aus der Entwicklung einer Medienverwaltung für eine Medienbibliothek ergeben. Die Stunde ist methodisch an den informatischen Vorstellungen der Schüler orientiert und stellt das informatische Problem und die Quellcodes in den Mittelpunkt.

Prinzipielle Vorgehensweise: Den Schülern wird ein Problem präsentiert, für welches Vererbung die optimale Lösung ist. Die Vererbungslösung selbst können die Schüler nicht finden, da sie Vererbung nicht kennen. Stattdessen analysieren, verstehen und bewerten die Schüler eine vorgegebene Lösung und lernen dadurch das Vererbungskonzept kennen.

Alternativ wurde erwogen, sich dem Thema über die Verhältnisse von realen Objekten der Welt zu nähern (siehe Sachanalyse), allerdings führt gerade diese Annäherung schnell in die Irre, weil Objekte konkret ausgeprägte Eigenschaften besitzen, in der informatischen Vererbung aber abstrakte Kategorien vererbt werden. Die hohe Abstraktion des Vererbungskonzepts macht auch eine Annäherung auf relativ abstrakter Ebene sinnvoll.

Einstieg:

Der Einstieg knüpft an die Modellierung einer Medienbibliothek aus der letzten Stunde an. Die Notwendigkeit der Ausdifferenzierung der Klasse Medium soll den Schülern klar werden. Dazu wird eine neue Eigenschaft *betriebssystem*¹ für erforderlich erklärt, welche nur sinnvoll im Zusammenhang mit CD-ROMS ist, nicht aber mit allen Medien. Es werden Lösungsvorschläge diskutiert. Hier ist unter Umständen geeignete Lenkung gefordert um Fehlvorschläge notfalls adäquat zu problematisieren.

Sollten die Schüler keine Vorschläge bringen, setze ich den Impuls, der zum Problem führt: *„Ich behaupte, dass man das Attribut „betriebssystem“ zu Medium hinzufügen kann.“*

Nach der Diskussion über das Problem wird die Erarbeitungsphase eingeleitet.

Möglicherweise geschieht eine Vorwegnahme des Vererbungsbegriffs durch einen der leistungsstarken Schüler in dieser Phase. Mögliche Antworten sind: *„Danke, Was haben die anderen für Ideen?“* bzw. *„Schön, dass du das schon siehst, ich möchte aber noch nicht darauf eingehen, sondern euch auffordern zuerst dieses Arbeitsblatt zu bearbeiten, und zu schauen, wie sich dort dem Problem genähert wird.“*

Erarbeitung:

Die Erarbeitung der Vererbung erfolgt hier deduktiv. Auf dem Arbeitsblatt findet sich eine Lösung im Stile der Vererbung. Das eigentliche Problem besteht jetzt weniger darin, wie die Lösung ist, sondern darin, was der präsentierte Code macht und was das mit dem Problem zu tun hat. Lernpsychologisch ist dieser Schritt problematisch, es wird versucht, Frust zu vermeiden, indem die Aufgabe nicht als „Lösung“ bezeichnet wird. Trotzdem ist die neue Frage für die Schüler durchaus interessant und motivierend, da die präsentierte Lösung mit dem ihnen bekannten Vorwissen zu kollidieren scheint. Im Detail sind das Zuweisungen, die an scheinbar nicht existente Attribute vorgenommen werden und eine kognitive Dissonanz hervorrufen (könnten). Mögliche Fragen der Schüler wären *„Ist das hier nicht falsch...?“*

Eine kooperative Sozialform ist sehr günstig, um solche auftretenden Fragen zu klären. Ich entscheide mich für drei Dreiergruppen und eine Zweiergruppe, wobei die Dreiergruppen keinen der beiden sehr leistungsstarken Schüler enthalten sollen, um

¹ Diese Eigenschaft soll ausdrücken, auf welchem System die auf der CD-ROM befindliche Software läuft.

den Schülern mit weniger Vorwissen die Selbsterarbeitung zu ermöglichen. Das bedeutet, dass die starke Zweiergruppe im Fall von feststellbarer Unterforderung einen Zusatzauftrag bekommen kann, nämlich „*Formuliert in einem Satz, was das Prinzip ausmacht, welches hier zum Einsatz kommt.*“

Es scheint mir nicht unbedingt notwendig, das Wort *extends* vorab zu klären. Dass es eine Beziehung zwischen den zwei Klassen ausdrückt, ergibt sich auch schon aus seiner Position; welche Beziehung das ist, kann annähernd aus der Semantik von CDROM gefolgert werden. Außerdem ist auch die Übersetzung „erweitert“ nicht unbedingt selbsterklärend.

Das Arbeitsblatt enthält, um Irritationen zu vermeiden, die Klasse Medium nicht im Quellcode, da verschiedene Klassen in Java auch nicht in derselben Datei vorliegen oder im selben Fenster zu sehen sind. Deswegen wurde die Diagrammform für Medium gewählt.

Ich werde den Verlauf der Gruppenarbeit beobachten und werde Gruppen, die große Probleme mit dem Arbeitsblatt haben, ein Hinweisblatt (siehe Anhang) mit Hilfsimpulsen zur Verfügung stellen, welches aber nur bei deutlicher Überforderung der ganzen Gruppe vom mir ausgegeben wird.

Auswertung

Ein Schüler einer Gruppe präsentiert am OHP, was seine Gruppe herausgefunden hat. Wenig sinnvoll scheint es, weitere Ergebnisse am OHP vorführen zu lassen, die mündliche Ergänzung wird hier vorgezogen. Es ist zu garantieren, dass die Schüler erkennen, dass die in der Methode verwendeten Attribute zur Klasse CDROM gehören. Das wird unterstützt durch die Formulierung der Ausschriften der Methode („*Diese CD-ROM hat die Mediennummer...*“) Weiterhin sind in der Verlaufsplanung entsprechende Impulse vorgesehen. Um die Beziehung zwischen den Klassen Medium und CDROM zu benennen, werden die Schüler aufgefordert, in der von ihnen erkannten Vererbungsanweisung „CDROM extends Medium“ das Schlüsselwort *extends* durch ein deutsches Verb oder eine Wortgruppe sinnvoll zu ersetzen. Des Weiteren kann eine Würdigung der Zusatzaufgabe der starken Zweiergruppe erfolgen, deren Aufgabe eine Beschreibung des Vererbungsprinzips darstellte. Der Vererbungs begriff wird von Schülern oder vom Lehrer genannt.

Wenn es in dieser Phase zu Verständnisproblemen kommt, sollen weitere Hilfsimpulse den Schülern das Verständnis erleichtern.

Sicherung

Den Schülern wird ein die Vererbung abstrakt darstellendes Schema präsentiert, in welches die beiden Klassen Medium und CDROM von einem Schüler eingeordnet werden sollen. Zusätzlich sollen die Besonderheiten des Schemas (Strichpunktlinien, Pfeile) erklärt und beschriftet werden.

Methodisch hat sich die Erklärung eines Schemas oft als fruchtbar erwiesen um Wissen zu sichern. Das Schema ermöglicht Vererbung zu abstrahieren und vom Medium/CDROM- Beispiel zu lösen. Dabei sollen die Schüler jetzt eigene Begriffe für Oberklasse und Unterklasse finden, die Fachbegriffe Superklasse und Subklasse werden dann vom Lehrer genannt.

Anwendungsphase

Durch die Entscheidung die Vererbung am Quellcode zu erarbeiten ergibt sich die Rechnerarbeit als funktionale Konsequenz. Hier können die Schüler die Theorie direkt überprüfen. Die Rechner sind soweit vorbereitet, dass nur noch der Monitor an-

geschaltet werden muss. Trotzdem kann die Rechnerarbeit nur stattfinden, wenn mindestens noch 15 Minuten Unterrichtszeit vorhanden sind. Alternativ ist vorgesehen sich auf Anwendungsaufgabe 1 zu beschränken, welche auf dem Papier zu lösen ist. Aufgabe 1 ist hinreichend um eine Anwendung zu gewährleisten, veranschaulicht wird das Ergebnis (Quellcode) auf einer Folie.

In jedem Fall wird die Gruppenarbeitsform und -sitzanordnung aufgelöst. Wegen der beschränkten Anzahl Rechner arbeiten einige Schüler in Partnerarbeit, dies sind sinnvoller Weise die tendenziell leistungsschwächeren Schüler. Dies ergibt sich in gleichem Maße für die eventuell stattfindende schriftliche Umsetzung bei Zeitproblemen.

Folgende Aufgabe wird gestellt:

Aufgabe

1. Füge unter Verwendung des Prinzips der Vererbung eine Klasse `SpielCDROM` zu unserer Medienbibliothek hinzu. Deren spezifische Attribute sollen *Spielgenre* und *Altersbegrenzung* sein.
2. Erstelle eine Methode `fuellen()`, die allen Attributen von `SpielCDROM` einen Wert zuweist.
3. Überprüfe deine Methode an einem Objekt.

Zeit: 10 Minuten

Die Aufgabe hat einen erhöhten Schwierigkeitsgrad, der darin besteht, eine neue Klasse von einer Klasse abzuleiten, die bereits geerbte Attribute enthält und zu erkennen, dass alle geerbten Attribute weitervererbt werden.

Die hinzuzufügende Klasse `SpielCDROM` soll von `CDROM` abgeleitet sein und zeichnet sich durch Attribute wie *Spielgenre* und *Alterbegrenzung* aus, welche vorgegeben werden.

Die Auswertung der Rechnerarbeit erfolgt durch kurze Präsentation des erstellten Objekts in BlueJ am Beamer. Zusätzlich soll die erstellte Klasse auf der Vererbungsfolie eingeordnet werden (Transfer).

Eine Hausaufgabe sichert die Anwendung der Vererbung.

7. Verlaufsplanung

Phase	Lernziel	Lehrertätigkeit	erwartete Schülertätigkeit	Medien/Sozialform
Einstieg 10.00-10.05 5 min	FZ1: Die S. erkennen, dass eine Klasse Medium nicht alle Eigenschaften ihres Spezialfalls CD-Rom haben kann, indem sie das Hinzufügen eines weiteren Attributs diskutieren	<p><i>L: Zur Erstellung einer Medienbibliothek haben wir gestern Attribute und Methoden zusammengetragen.</i></p> <p><u><i>Nennt Medien, die unsere Medienbibliothek enthalten kann.</i></u></p> <p><i>Ich habe an der Tafel eine stark reduzierte Version von der Klasse Medium angebracht.</i></p> <p><i>L: Wir verleihen auch CD-ROMs mit Software und es gibt Kunden, die wünschen, dass man bei den CD-ROMs erkennen kann, für welches Betriebssystem die Software ist. <u>Wie können wir das umsetzen?</u></i> <i>Was ist der Nachteil dieser Lösung?</i></p> <p><i>HI: Ich behaupte, dass man das Attribut „betriebssystem“ zu Medium hinzufügen kann.</i></p>	<p><i>S: DVD, Buch, CD-Rom</i></p> <p><i>S:</i></p> <ol style="list-style-type: none"> <i>Das geht nicht.</i> <i>füge Attribut an Klasse Medium an. S. diskutieren Lösung „gilt dann nicht für Buch etc.“</i> <i>Erstelle neue Klasse CD-ROM und kopiere alles rein S. diskutieren Lösung „Aufwendig, alles doppelt etc.)</i> <i>Medium beseitigen und ausdifferenzierte Klassen erstellen.</i> <i>Erstelle neue Klasse CD-ROM und nutze Vererbung (⊗ Vorwegnahme)</i> 	Tafel: <pre>class Medium{ int mediennummer; String titel; int anzahl_ausleihen; }</pre> GUG
Erarbeitung 10.06-10.16 10 min	FZ2: Die S. kennen die Syntax und Semantik von ererbenden Klassen, indem sie einen Quellcode nachvollziehen und die dortigen Attribute als übernommen interpretieren.	<p><i>Wir wollen uns eine andere Möglichkeit erarbeiten. Löst dazu in Gruppen das Arbeitsblatt. Ihr habt 10 Minuten Zeit.</i></p> <p><i>L. geht herum und beobachtet Gruppentätigkeit, gibt notfalls Hilfefzettel bzw. Zusatzaufgabe aus.</i></p>	Schüler bearbeiten Arbeitsblatt.	Gruppenarbeit Arbeitsblatt

<p>Auswertung</p> <p>10.16-10.24 8 min</p>	<p>FZ3: Die S. wissen, dass die Attribute der Oberklasse auch zur Unterklasse gehören, indem sie sagen, dass die Methode ausgeben() Eigenschaften von CDROM ausgibt.</p> <p>FZ4: Die S. können das Verhältnis von Unterklasse und Oberklasse beschreiben, indem sie für das Schlüsselwort <i>extends</i> einen adäquaten deutschen Ersatz finden.</p>	<p>Bittet einen S. an den OHP. <i>Stell die Ergebnisse deiner Gruppe vor.</i></p> <p>Was haben die anderen Gruppen diskutiert? Bitte ergänzt. HI: Was macht die Methode ausgeben? HI: Über wen gibt die Methode ausgeben Informationen aus? HI: Wie kann man in einem Satz zusammenfassen, was die Methode ausgibt? HI: Woher hat die Methode diese Informationen?</p> <p><u>Wie ist das möglich, da die Eigenschaften/Attribute zu Beginn der Klasse deklariert sind und sich dort nur <i>betriebsystem</i> befindet?</u></p> <p>Wo wird dies angewiesen?</p> <p><u>Ersetzt <i>extends</i> durch ein euch passend scheinendes deutsches Verb</u> oder eine Wortgruppe, das das Verhältnis von CDROM und Medium ausdrückt (es muss keine Übersetzung sein):</p> <p>Was bedeutet <i>extends</i> übersetzt? Wie ist das zu verstehen?</p> <p>Was hat die Gruppe, die ich bat, das Prinzip zu formulieren, aufgeschrieben?</p> <p>L. nennt „Vererbung“. legt Folie auf Dieses Prinzip nennt man Vererbung.</p>	<p>Schüler stellt vor, zeigt und markiert auf Folie, z.B.</p> <p>„ja das sind zwei Klassen, die eine besteht aus den Attributen ... und die andere aus einem Attribut und einer Methode. Es gibt hier hinter dem Klassennamen noch <i>extends Medium</i>, da weiß ich nicht, was das heißen soll. Die Methode <i>ausgeben()</i> schreibt Attributwerte aus von der Klasse <i>Medium</i>.“ (minimale Lösung, erfordert Ergänzung bzw. Steuerung)</p> <p>Schüler sagen: Die Methode gibt die Eigenschaften von CDROM aus!</p> <p>S: Sie erhält Attribute von <i>Medium</i>.</p> <p>S: <i>CDROM extends Medium</i></p> <p>S: -holt sich Attribute von -bekommt von -erbt von -erweitert</p> <p>S:Bei diesem Prinzip bekommt/erbt die Klasse <i>CDROM</i> mithilfe von <i>extends</i> die Attribute von <i>Medium</i>.</p>	<p>Folie</p> <p>GUG</p>
---	---	---	---	-------------------------

<p>Sicherung</p> <p>10.25-10.30 5 min</p>	<p>FZ5: Die S. können ein Vererbungsverhältnis abstrakt erkennen und beschreiben, indem sie eine schematische Darstellung interpretieren und erklären.</p>	<p><i>Ich habe eine Grafik im Internet gefunden, die darstellen will, was wir herausgefunden haben, leider ohne jegliche Erläuterung oder Beschriftung. Ausgabe der Arbeitsbögen II.</i></p> <p><i>Zeigt, wo sich die Klassen von unserem Arbeitsblatt in diesem Schema wieder finden.</i></p> <p><i>Erklärt die Bedeutung der gestrichelten Kästchen und die Pfeile.</i></p> <p><i>(Was könnten die anderen Rechtecke bedeuten?)</i></p> <p>Folie zurückklappen. <i>Hat jemand einen Vorschlag, wie wir die Klassen <u>aufgrund ihrer Beziehung</u> bezeichnen könnten?</i></p> <p>Oberklasse/Superklasse – Unterklasse/Subklasse <i>Bitte wendet euer neues Wissen am Rechner mit folgender Aufgabe an. Ihr habt 10 Minuten Zeit.</i></p>	<p>S. schreiben „Klasse Medium“ und „CDROM“ auf Folie (ggf. einschließlich Attribute)</p>	<p>(Klapp)-Folie</p>
<p>Anwendung</p> <p>10.30-10.45 15 min</p>	<p>FZ6: Die S. wenden das Vererbungsprinzip an, indem Sie (am Rechner) eine Subklasse erstellen, diese auf die richtige Superklasse beziehen und alle vorhandenen Attribute ansprechen. (fakultativ)</p>	<p>teilt Aufgabenzettel aus</p> <p><i>Von welcher Klasse wurde geerbt? Wo auf unserer Folie ordnet ihr die Klasse SpieleCDROM ein?</i> (legt Folie auf) <i>Wie viele Zuweisungen müssen verwendet worden sein?</i></p> <p>L erteilt HA: Erstellt zu Hause für diese Medienbibliothek eine Klasse <i>AnwendungsCDROM</i> und eine Klasse <i>Buch</i>. Schickt mir alle notwendigen Java-Klassen per E-Mail.</p>	<p>bearbeiten Aufgabe am Rechner</p> <p>S.: Es werden 6 Zuweisungen verwendet.</p>	<p>Rechner. teilweise Partnerarbeit</p> <p>Folie</p>

8.Literatur

Barnes,D.; Kölling, M.: Objektorientierte Programmierung mit Java. Eine praxisnahe Einführung mit BlueJ. Pearson 2003.

Balzert, Helmut: Lehrbuch Grundlagen der Informatik. Spektrum 1999.

Partl, Hubert: Der bhv-Voach Java. Bonn 2002.

9. Anhang

- Arbeitsbögen
- Gruppenzuordnung
- diagnostische Matrix
- Sitzplan

.....
Unterschrift des Prüfungskandidaten

Didaktische Gruppenzuordnung

Gruppe 1: Benjamin, Markus, Patrick

Gruppe2: Ben, Felix

Gruppe3 : Tobias, Franz, Steffen

Gruppe4: Jabin, Paul, Ludwig

Arbeitsblatt



Aufgabe

Untersucht in der Gruppe den dargestellten Quellcode.

1. Beschreibt, wie er aufgebaut ist.
2. Markiert Bestandteile, die unbekannt bzw. neu für euch sind.
3. Diskutiert eventuelle Besonderheiten. Stellt Vermutungen über Verhalten und wahrscheinliche Funktionsweise des Programms an.
4. Bereitet euch auf eine Präsentation eurer Ergebnisse vor.

Zeit: 10 Minuten

Diagramm von Klasse Medium

signatur
titel
anzahl_ausleihen

```
class CDROM extends Medium{

    String plattform;

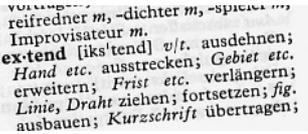
    void ausgeben(){
        System.out.println("Diese CD-ROM hat die Signatur "
            +signatur);

        System.out.println("Diese CD-ROM hat den Titel "+titel);

        System.out.println("Diese CD-ROM wurde bisher " +
            anzahl_ausleihen + " Mal ausgeliehen.");

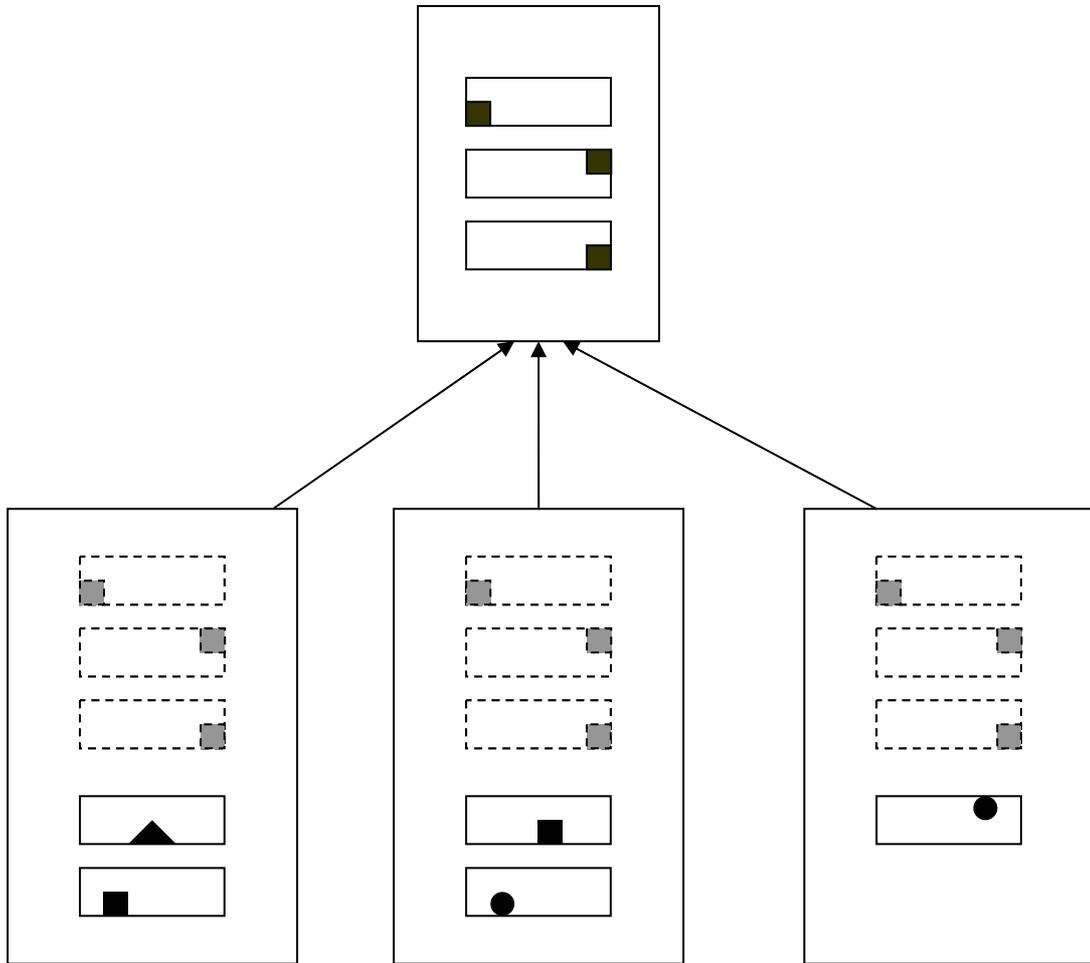
        System.out.println("Die auf dieser CD-ROM befindliche
            Software läuft auf " + plattform);
    }
}
```

Hilfreiche Hinweise zum Arbeitsblatt

1. 

reifredner *m*, -dichter *m*, -spieler *m*,
Improvisateur *m*.
ex·tend [iks'tend] *v/t.* ausdehnen;
Hand etc. ausstrecken; *Gebiet etc.*
erweitern; *Frist etc.* verlängern;
Linie, Draht ziehen; fortsetzen; *fig.*
ausbauen; *Kurzschrift* übertragen;
2. In Klasse CDROM wird auf Elemente zugegriffen, die darin gar nicht existieren. Welche?
3. Was könnte der Grund dafür sein, dass der Zugriff trotzdem funktioniert?

VERERBUNG



Diagnostische Matrix der Lerngruppe WPU10 im Fach Informatik

sachstruktureller Entwicklungsstand	Mitarbeit (Aktion)	Beteiligung (Interaktion)																																																																		
Kenntnisse Fertigkeiten	Güte Konstanz	Partizipationsneigung Verlaufsdienlichkeit																																																																		
+ umfangreich und sicher + o solide u. verfügbar o - identifizierbare und benennbare Lücken -	+ erfolgreich und sorgfältig + o bedarf didaktischer Rückkoppelung o - benötigt Lernhilfen -	+ immer hoch + o gemäßigt oder schwankend o - gering -																																																																		
+ sichere und zügige Ausführung + + o solide Ausführung mit Unebenheiten o - unsichere Ausführung oder benennbare Mängel -	+ selbstgesteuert altersgemäß + o bedarf der Kontrolle o - Animation notwendig -	+ immer verlaufsgerecht + o in der Regel verlaufsdienlich o - häufig hinderlich -																																																																		
Felix <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><td style="text-align: center;">+</td><td style="text-align: center;">+</td><td style="text-align: center;">+</td></tr> <tr><td style="text-align: center;">+</td><td style="text-align: center;">+</td><td style="text-align: center;">+</td></tr> </table> Franz <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><td style="text-align: center;">o-</td><td style="text-align: center;">o-</td><td style="text-align: center;">-</td></tr> <tr><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">o</td></tr> </table> Paul <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><td style="text-align: center;">o+</td><td style="text-align: center;">o</td><td style="text-align: center;">o</td></tr> <tr><td style="text-align: center;">o</td><td style="text-align: center;">o</td><td style="text-align: center;">+</td></tr> </table> Steffen <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><td style="text-align: center;">+</td><td style="text-align: center;">+</td><td style="text-align: center;">o</td></tr> <tr><td style="text-align: center;">+</td><td style="text-align: center;">o</td><td style="text-align: center;">+</td></tr> </table>	+	+	+	+	+	+	o-	o-	-	-	-	o	o+	o	o	o	o	+	+	+	o	+	o	+	Ben <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><td style="text-align: center;">+</td><td style="text-align: center;">+</td><td style="text-align: center;">+</td></tr> <tr><td style="text-align: center;">+</td><td style="text-align: center;">+</td><td style="text-align: center;">+</td></tr> </table> Tobias <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><td style="text-align: center;">o-</td><td style="text-align: center;">o</td><td style="text-align: center;">o</td></tr> <tr><td style="text-align: center;">o-</td><td style="text-align: center;">o</td><td style="text-align: center;">+</td></tr> </table> Benjamin <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><td style="text-align: center;">+</td><td style="text-align: center;">o+</td><td style="text-align: center;">o</td></tr> <tr><td style="text-align: center;">o+</td><td style="text-align: center;">o</td><td style="text-align: center;">+</td></tr> </table> Patrick <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><td style="text-align: center;">o</td><td style="text-align: center;">o</td><td style="text-align: center;">o-</td></tr> <tr><td style="text-align: center;">-</td><td style="text-align: center;">o</td><td style="text-align: center;">+</td></tr> </table>	+	+	+	+	+	+	o-	o	o	o-	o	+	+	o+	o	o+	o	+	o	o	o-	-	o	+	Marcus <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><td style="text-align: center;">o-</td><td style="text-align: center;">o</td><td style="text-align: center;">o</td></tr> <tr><td style="text-align: center;">o</td><td style="text-align: center;">o</td><td style="text-align: center;">o</td></tr> </table> Jabin <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><td style="text-align: center;">-</td><td style="text-align: center;">-</td><td style="text-align: center;">o</td></tr> <tr><td style="text-align: center;">-</td><td style="text-align: center;">o</td><td style="text-align: center;">-</td></tr> </table> Ludwig <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><td style="text-align: center;">+</td><td style="text-align: center;">+</td><td style="text-align: center;">o+</td></tr> <tr><td style="text-align: center;">o</td><td style="text-align: center;">o</td><td style="text-align: center;">o</td></tr> </table>	o-	o	o	o	o	o	-	-	o	-	o	-	+	+	o+	o	o	o
+	+	+																																																																		
+	+	+																																																																		
o-	o-	-																																																																		
-	-	o																																																																		
o+	o	o																																																																		
o	o	+																																																																		
+	+	o																																																																		
+	o	+																																																																		
+	+	+																																																																		
+	+	+																																																																		
o-	o	o																																																																		
o-	o	+																																																																		
+	o+	o																																																																		
o+	o	+																																																																		
o	o	o-																																																																		
-	o	+																																																																		
o-	o	o																																																																		
o	o	o																																																																		
-	-	o																																																																		
-	o	-																																																																		
+	+	o+																																																																		
o	o	o																																																																		