# Auftrag 2: Vererbungshierarchie implementieren

Wie Sie bereits in der vorherigen Lektion gesehen haben, ist eines der wichtigsten Ziele in der objektorientierten Programmierung, die Vermeidung von redundantem Code, wie er typischerweise durch *Copy-Paste* entsteht. Attribute und Methoden sollen genau einmal deklariert und definiert werden und nicht in mehreren Klassen auftauchen. Ein Hilfsmittel hierzu sind *Klassenhierarchien*. Durch *Vererbung* können Attribute und Methoden in *abgeleiteten* Klassen verfügbar gemacht werden. Dieses Prinzip haben Sie kennengelernt, nun wenden Sie es auf Ebene Java-Code an.

Lesen Sie auch im Buch das **Kapitel 3.5**

## Aufgabe 1: Vererbung und Referenzen

*Einleitung:*

Im Rahmen der Vererbung muss man bei Referenzen von Basisklasse bzw. abgeleiteter Klasse einige Besonderheiten beachten.

*Aufgabe:*

Im Screencast „Vererbung und Referenzen.mp4“ in den Ressourcen, werden diese Besonderheiten beim Einsatz von Referenzen erklärt. Schauen Sie sich diesen an, und beantworten Sie die Fragen zu folgendem Java-Programm:

Personen.java

|  |
| --- |
| **class** Person {  String name;  String vorname;  **int** personalNummer;  }  **class** Chef **extends** Person {  String abteilung;  }  **class** Fachangestellter **extends** Person {  Chef vorgesetzter;  }  **public** **class** Personen {  **public** **static** **void** main (String args[]) {  Fachangestellter personal1 = **new** Fachangestellter();  Fachangestellter personal2 = **new** Fachangestellter();  Chef personal3 = **new** Chef();  Person personal4 = **new** Person();  personal3.name="Sattler";  personal3.vorname="Beatrice";  personal3.abteilung="Verkauf";  personal1.name="Klein";  personal1.vorname="Thomas";  personal1.vorgesetzter=personal3; //\*1  personal4 = personal1; //\*2  personal4.vorgesetzter = personal3; //\*3  personal1 = personal4; //\*4  personal1 = (Fachangestellter) personal4; //\*5  }  } |

In vorstehendem Code sind die letzten 5 Zeilen mit \*1 bis \*5 gekennzeichnet. Überlegen Sie für jede dieser Zeilen, ob diese Programmanweisungen ohne Fehlermeldung, bzw. ohne Laufzeitfehler ablaufen können.

|  |  |
| --- | --- |
| \*1 |  |
| \*2 |  |
| \*3 |  |
| \*4 |  |
| \*5 |  |

Kontrollieren Sie Ihre Antworten, indem Sie den Code in ein Java-Projekt in eclipse kopieren.

## Aufgabe 2: Vererbung und Konstruktoren

*Einleitung:*

K 3.1

*Konstruktoren* sind spezielle „Methoden“, welche zur *Initialisierung* von Objekten aufgerufen werden. Wie alle Methoden, können auch Konstruktoren *Parameter* haben, über welche beim Aufruf Werte mitegegben werden können. Das wird hier ausgenützt, um die Instanzvariablen der Objekte bei der Erzeugung mit den gewünschten Werten initialisieren zu können.

Schlagen Sie bei Bedarf im Buch das Kapitel 3.3, Konstruktoren, nochmals nach und lesen Sie **Seite 50** zum Thema Konstruktoren.

*Aufgabe:*

Ergänzen Sie jetzt die Klassen Person, Chef und Fachangestellter mit je einem passenden Konstruktor. Dieser soll die Instanzvariablen der jeweiligen Klasse mit den durch die Parameter erhaltenen Werte initialisieren.

Setzen Sie jetzt die Instanzvariablen der Klassen auf private, respektive, wo nötig, auf protected.

Orientieren Sie sich an dem Beispiel des Screencasts " Vererbung und Konstruktoren.mp4".

## Aufgabe 3: Vererbung und Speicherdiagramm

*Einleitung:*

K 2.2

Zur Veranschaulichung von Referenzen und Arrays oder Objekten wurde bereits mehrfach die Situation im Speicher aufgezeichnet. Genau darum geht es hier.

*Aufgabe:*

Erweitern Sie Ihr Programm Personen.java aus Aufgabe 1 um die folgende Klasse:

**class** Firma {

**private** Person[] mitarbeiter;

**public** Firma (Person[] p) {

mitarbeiter = p;

}

}

Ersetzen Sie nun in Ihrer Klasse Personen im Programm Personen.java aus Aufgabe 1 die Anweisungen der Main-Methode durch die folgenden:

Firma f = **new** Firma(

**new** Person[] {**new** Chef("Sattler", "Beatrice",25, "Verkauf")} ); // <==

Zeichnen Sie die Situation im Speicher nach dem Ausführen der mit **// <==** bezeichneten Zeile.

Halten Sie sich an die Notation auf der verteilten Vorlage zu Speicherdiagrammen (R226A\_Speicherdiagramme.doc, Lektion 6 im Modul 226A).

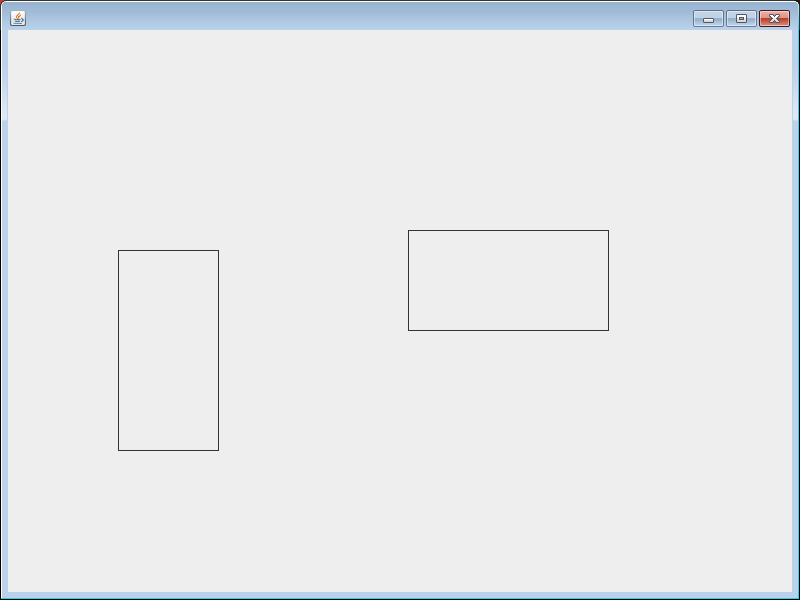
Beschriften Sie alle Speicherzellen mit Name, Typ und Inhalt.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Aufgabe 4: Grafik-Editor Version 1

*Einleitung:*

Wie bereits im ersten Arbeitsblatt erwähnt, werden die neuen Inhalte während des gesamten Semesters mit Hilfe des Programms „Grafikeditor“ erarbeitet. Es ist daher wichtig, dass Sie bereits zum jetzigen Zeitpunkt beginnen, an diesem Programm zu arbeiten.

Wenden Sie nun also die in diesem Arbeitsblatt erworbenen Kenntnisse in einer ersten Version des Programms Grafikeditor an.

Version 1:

In einem Programm erstellte Rechteck-, Kreis- und Linien-Objekte können auf dem Bildschirm in einem Fenster angezeigt werden. Für die grafische Ausgabe im Fenster steht in den Ressourcen die Datei "Display.java" zur Verfügung.

*Aufgabe:*

Eröffnen Sie ein neues Projekt.

1. Schreiben Sie je eine Klasse für die grafischen Figuren, Rechteck, Linie und Kreis.

Überlegen Sie, welche Informationen Sie brauchen, wenn Sie ein Rechteck zeichnen sollen. *Deklarieren* Sie in der Klasse Rechteck die nötigen *Instanzvariablen* (vorerst ohne *Modifizierer*). Tun Sie dies anschliessend für die anderen Klassen.

Haben Sie identische Deklarationen in mehreren Klassen? Führen Sie die *Superklasse* Figur ein und verschieben Sie redundante Deklarationen dorthin.

Schreiben Sie schliesslich eine Klasse Main mit main-Methode. Diese soll einige Rechteck-, Linien- und Kreisobjekte erzeugen.

1. Erweitern Sie jetzt die Klassen Figur, Rechteck, Linie und Kreis um je einen passenden Konstruktor. Dieser soll die Instanzvariablen der jeweiligen Klasse mit den durch die Parameter erhaltenen Werte initialisieren.

Setzen Sie jetzt die Instanzvariablen der Klassen auf private, respektive, wo nötig, auf protected.

1. Integrieren Sie die Klasse Display aus der Datei "Display.java" in Ihr Projekt. Studieren Sie den Code in dieser Datei. Wenn Sie sich die Methode zeichneFiguren(Graphics g) ansehen, so sehen Sie, dass dort das Zeichnen von Rechtecken implementiert ist.

Deklarieren Sie in Ihrer Main-Klasse eine Klassenvariable display:

**private** **static** **final** Display ***display*** = **new** Display();

Lassen Sie nun die Rechtecke im Fenster ausgeben.

1. Damit auch Ihre anderen Figuren gezeichnet werden können, müssen Sie die Methode zeichneFiguren(Graphics g) in der Klasse Display noch entsprechend ergänzen.

## Vertiefungs- und Ergänzungsaufgaben

Die folgenden Aufgaben dienen Ihrer individuellen Vertiefung. Sie können diese nach Belieben bearbeiten und dadurch auch Zusatzkompetenzen erlangen. Die Aufgaben sind bewusst offen formuliert und Sie werden für die Lösung im Web recherchieren müssen.

1. Implementieren Sie weitere Figurklassen, z.B. Text, Bogen, Ellipse, etc.
2. Ergänzen Sie Ihre Figur-Klassen mit weiteren Eigenschaften, wie Füllung, Linienfarbe, Liniendicke, Linienart, etc. Überlegen Sie sich dabei, auf welcher Stufe der Vererbungshierarchie diese Eigenschaften angesiedelt werden müssen. Sie werden dabei die Klasse Graphics2D benötigen. Im Web (Java-API-Doc und Stackoverflow) finden Sie weitere Informationen.
3. Ein Grafikeditor bietet normalerweise auch die Möglichkeit, einzelne Figuren zu einer Gruppe zusammenzufassen. Dies ist ein typischer Fall, wo das *Composite-Design-Pattern* zur Anwendung kommt. Sie finden im Web mehr Informationen dazu. Implementieren Sie also die Klasse Gruppe als Composite-Klasse. Ein Gruppenobjekt, das „Ganze“, enthält dann zur Laufzeit einzelne Figur-Objekte als Teile (die gute alte Aggregation aus Modul 226-1).

K 1.5