

Übungsblatt 1

Besprechung in der Vorlesung am 21.03.2006

1 Beim objekt-orientierten Bäcker

Beim Bäckermeister Krause kann man verschiedene Backwaren kaufen. Herr Krause steht hinter einer Theke, auf der eine Kasse steht. Herr Krause ist ein moderner Bäcker und hat bereits eine Scannerkasse. Das Warenangebot wird den Kunden in der Theke präsentiert. Die Kunden kaufen Brötchen, Brot, Kuchen und kleine Teilchen. Herr Krause ist so nett und verpackt die gewünschten Waren passend in verschieden große Tüten. Jeder Kunde bekommt neben der Tüte einen Kassenbon, auf dem die Waren und ihr Preis verzeichnet sind. Der Bäckerladen von Herrn Krause öffnet morgens schon um 6 Uhr, um auch hungrige Erstsemester, die von einer Semesterparty kommen, mit warmen Brötchen versorgen zu können.

1. Identifizieren Sie Objekte und begründen Sie Ihre Wahl kurz anhand der in der Vorlesung genannten Prinzipien.
2. Gruppieren Sie die Objekte zu Klassen und stellen Sie sie in einem UML-Klassendiagramm dar.
3. Geben Sie für die Klassen die Schnittstellen an. Identifizieren Sie dabei Attribute und Methoden.
4. Beschreiben Sie die vielfältigen Beziehungen zwischen den gefundenen Klassen mit Hilfe der in der Vorlesung vorgestellten Modellierungstechniken.

2 Hello World!

In den folgenden Übungen werden Sie kleine Programme in Java schreiben. Dies soll nun geübt werden, damit Sie an den folgenden Tagen gleich mit dem eigentlichen Programmieren loslegen können. Führen Sie die folgenden Schritte dazu einmal nur mit den Mitteln des JDK auf der Kommandozeile, und einmal mit den Mitteln einer Entwicklungsumgebung (z.B. BlueJ) durch.

1. Geben Sie mit Hilfe eines geeigneten Editors das HelloWorld-Programm aus der Vorlesung ein und speichern Sie es.
2. Kompilieren Sie das Programm.
3. Führen Sie das Programm aus.

Tipps für den Umgang mit dem JDK auf der Kommandozeile:**Kompilieren:**

Fehler: „Der Befehl "javac" ist entweder falsch geschrieben oder konnte nicht gefunden werden.“

Mögliche Ursache: Das System weiß nicht, wo es nach dem Programm javac suchen soll und kann es in den Verzeichnissen, wo es üblicherweise nach Programmen sucht, nicht finden.

Mögliche Abhilfe:

- Geben Sie den vollständigen Pfad des Compilers an (z.B. `c:\Programme\Java\jdk1.5.0_06\bin\javac`) oder
- Fügen Sie das bin Verzeichnis zur Umgebungsvariablen Path hinzu: In der Eingabeaufforderung (gültig bis die Eingabeaufforderung beendet wird):
`set path=%path%;C:\programme\Java\jdk1.5.0_06\bin` (oder wo auch immer das JDK installiert ist.)

In Windows (dauerhaft):

Unter Einstellungen / Systemsteuerung / System / Erweitert / Umgebungsvariablen muss die Systemvariable Path bearbeitet werden. Zum schon vorhandenen Wert wird `;C:\programme\Java\jdk1.5.0_06\bin` (oder wo immer das JDK installiert ist) hinzugefügt. Dabei ist darauf zu achten, dass an den vorhandenen Einträgen nichts geändert werden sollte. Das Semikolon zwischen den vorhandenen und dem neu hinzugefügten Pfad nicht vergessen!

Fehler: Der Compiler startet zwar, erzeugt aber keine .class Datei und gibt auch keine Fehlermeldung aus.

Mögliche Ursache: Wenn sich die .java – Datei nicht im aktuellen Ordner befindet, kann es vorkommen, dass der Compiler die .class – Datei im Wurzelverzeichnis (z.B. `C:\`) erstellt.

Mögliche Abhilfen:

- Starten Sie den Compiler mit der Option `-verbose` . Dann werden zusätzliche Ausgaben während der Bearbeitung gemacht, unter anderem ist angegeben, wohin genau Dateien geschrieben werden (z.B. `[wrote c:\HelloWorld.class]`).
- Nutzen Sie die Option `-d` , um einen Ort der Zieldateien explizit anzugeben (z.B. `javac HelloWorld.java -verbose -d c:\temp\`)

Starten

Fehler: Exception in thread "main"
java.lang.NoClassDefFoundError: HelloWorld/class

Ursache: Der Aufruf der `.class` – Datei muss ohne Angabe des Suffix erfolgen (also z.B. `java HelloWorld` und **nicht** `java HelloWorld.class`)

Allgemeiner Hinweis: Achten Sie auf eine einheitliche Groß- und Kleinschreibung der Klassen und Dateinamen, denn `helloWorld` \neq `HelloWorld`!