

# 1. Übungsaufgabe

## Themen:

Datenabstraktion, Klassenzusammenhalt, Objekt-Kopplung

## Termine:

Ausgabe: 12.10.2011  
reguläre Abgabe: 19.10.2011, 13:45 Uhr  
nachträgliche Abgabe: 26.10.2011, 13:45 Uhr

## Abgabeverzeichnis:

Gruppe/Aufgabe1

## Programmaufruf:

java Test

## Grundlage:

Erstes Kapitel des [Skriptums](#)

## Aufgabe

### Welche Aufgabe zu lösen ist:

Schreiben Sie in Java den Kern eines Programms zur Anmeldung von Studierenden zu Lehrveranstaltungen. Aus Sicht eines Anwenders sollen folgende Vorgänge unterstützt werden:

- Anlegen einer neuen Lehrveranstaltung mit allen notwendigen Daten.
- Auflistung aller angebotenen Lehrveranstaltungen jeweils mit folgenden Daten:
  - eindeutige LVA-Nummer und Titel der Lehrveranstaltung,
  - frühesten und spätesten Anmeldetermin und spätesten Abmeldetermin,
  - Anzahl der bisher angemeldeten Studierenden.

Dabei soll aber keine Information darüber gegeben werden, wer zu welcher Lehrveranstaltung angemeldet ist.

- Anmeldung einer/eines Studierenden zu einer Lehrveranstaltung innerhalb des Anmeldezeitraums. Eine Studentin bzw. ein Student kann zu einer Lehrveranstaltung nur einmal angemeldet sein.
- Abmeldung einer/eines angemeldeten Studierenden von einer Lehrveranstaltung innerhalb des Abmeldezeitraums.
- Auflistung aller zu einer Lehrveranstaltung angemeldeten Studierenden mit Matrikelnummern und Namen.

Schreiben Sie nur den Kern eines entsprechenden Programms (ohne Benutzerschnittstelle) sowie ein Testprogramm. Dieser Kern soll die oben beschriebene Funktionalität aufweisen, aber keinerlei Eingabe von der Tastatur oder Ausgabe auf den Bildschirm (und auch nicht von/auf eine Datei) machen. Daher müssen alle benötigten bzw. erzeugten Daten über Parameter und Rückgabewerte von Methoden übergeben werden. Entsprechende Typen der Daten (z.B. *String* oder eigene Klassen und Interfaces) müssen Sie selbst finden. Die Daten im Anmeldesystem können bei Programmbeendigung verloren gehen. Es ist also nicht nötig, die Daten in einer Datei zu speichern.

Das Testprogramm soll mittels *java Test* von *Gruppe/Aufgabe1* aus aufrufbar sein und die oben beschriebene Funktionalität, vor allem aber auch Grenzfälle überprüfen. Die Tests sollen selbständig ohne Benutzerinteraktion ablaufen, sodass Aufrufer keine Testfälle auswählen oder Testdaten eintippen müssen. Alle Testdaten sollen im Testprogramm vorgegeben sein. Testergebnisse sollen aber (in nachvollziehbarer und verständlicher Form) am Bildschirm ausgegeben werden.

### **Wie die Aufgabe zu lösen ist:**

Der Programmcode Ihrer Lösung soll möglichst einfach sein und keine unnötige Funktionalität haben. Der Code soll auch wiederverwendbar sein, da die nächste Aufgabe voraussichtlich auf Teilen davon aufbaut. Vermeiden Sie jedoch Vorgriffe, das heißt, schreiben Sie *keine* Programmteile aufgrund der Vermutung, dass diese Teile in der nächsten Aufgabe verlangt sein könnten.

Lesen Sie das ganze erste Kapitel des Skriptums und achten Sie dabei darauf, wie Sie eine gute Qualität von Software und gute Wiederverwendbarkeit erreichen können. Achten Sie bei der Lösung der Aufgabe besonders auf Datenabstraktion: Alles, was kaum trennbar miteinander verbunden ist, soll in einem Objekt gekapselt sein, leicht voneinander trennbare Einheiten sollen zu verschiedenen Objekten gehören. Es soll in Ihrem Programm mehrere voneinander möglichst unabhängige Objekte geben. Auf Daten soll nur über dafür vorgesehene Methoden zugegriffen werden. Unnötige Zugriffe und unnötige Zugreifbarkeit von Daten und Methoden sind zu vermeiden. Achten Sie auf hohen Klassenzusammenhalt und schwache Objektkopplung.

Diese Aufgabe hilft auch den Tutoren bei der Einschätzung Ihrer Vorkenntnisse sowie der Zusammenarbeit in der Gruppe. Bitte sorgen Sie in Ihrem eigenen Interesse dafür, dass Sie diese Aufgabe in der Gruppe lösen (ohne Abschreiben) und jedes Gruppenmitglied in etwa in gleichem Maße mitarbeitet. Sonst könnten Sie bei einer Fehleinstufung wertvolle Zeit verlieren. Scheuen Sie sich bitte nicht, Ihren Tutor um Hilfe zu bitten, falls Sie bei der Lösung der Aufgabe Probleme haben oder keine brauchbare Zusammenarbeit in der Gruppe zustande kommt.

### **Warum die Aufgabe diese Form hat:**

Die Aufgabe kommt aus einem Anwendungsbereich, den Sie kennen. Der Umfang ist so gewählt, dass die Programmierung selbst nicht zu viel Zeit in Anspruch nimmt und dadurch Zeit für die interne Organisation der Gruppe (also zum Kennenlernen, Finden einer passenden Position in der Gruppe und Aufbau geeigneter Kommunikationsmöglichkeiten) und inhaltliche Diskussionen bleibt. Auf eine genaue Spezifikation der Aufgabenstellung wird aus folgenden Gründen bewusst verzichtet:

- Sie sollen in der Gruppe diskutieren, wie Sie die Aufgabe verstehen und welche Lösungswege geeignet erscheinen.
- Sie sollen sich von Anfang an daran gewöhnen, dass Aufgaben in der Regel nicht vollständig spezifiziert sind, aber trotzdem bestimmte Vorgaben eingehalten werden müssen.
- Sie sollen sich eine eigene brauchbare Faktorisierung überlegen und dabei von Qualitätsmerkmalen wie Datenkapselung, Klassenzusammenhalt und Objektkopplung (statt starrer Vorgaben) leiten lassen.
- Sie sollen auch die Verantwortung für die Korrektheit Ihrer Lösung (so wie Sie sie selbst verstehen) übernehmen, indem Sie entsprechende Tests durchführen.

## **Allgemeine Informationen zur Übung**

Folgende Informationen betreffen diese und auch alle weiteren Aufgabenstellungen.

### **Was Sie bei der Lösung der Aufgabe beachten sollen:**

Diese Aufgabenstellung lässt Ihnen (wie die meisten folgenden) viel Freiheit bei der Lösungsfindung, stellt in manchen Bereichen aber auch klare Anforderungen. Nützen Sie die Freiheit vor allem zur Vereinfachung der Lösung, vergewissern Sie sich aber, dass alle Anforderungen erfüllt sind.

Unter der Überschrift "Wie die Aufgabe zu lösen ist" finden Sie einerseits Hinweise darauf, wie Sie die Lösung der Aufgabe vereinfachen können und welche Fallen Sie umgehen sollen, erfahren andererseits aber auch, welche

Aspekte bei der Beurteilung als besonders wichtig betrachtet werden. Bei der ersten Aufgabe kommt es beispielsweise besonders auf Datenabstraktion, Klassenzusammenhalt, Objektkopplung und die Einfachheit der Lösung an. Das heißt, in späteren Aufgaben (die nicht mehr von Tutoren beurteilt werden) können Ihnen bei solchen Hinweisen auch für unnötig komplizierte oder umfangreiche Lösungen viele Punkte abgezogen werden, weil Sie sich nicht an die Vorgaben gehalten haben. "Fleißaufgaben" sind in diesem Fall also unerwünscht. Unterschiedliche Aufgaben haben unterschiedliche Schwerpunkte. Verlassen Sie sich nicht darauf, dass die nächste Aufgabe nach demselben Schema beurteilt wird wie die vorangegangene, sondern richten Sie sich nach der jeweiligen Aufgabenstellung.

Ein häufiger Fehler besteht darin, eine Aufgabe nur nach Gefühl zu lösen ohne zu verstehen, worauf es ankommt. Meist bezieht sich die Aufgabe auf ein Thema, das kurz zuvor in der Vorlesung behandelt wurde. Versuchen Sie, eine Beziehung zwischen der Aufgabenstellung und dem Vorlesungsstoff herzustellen. Achten Sie besonders darauf, Fachbegriffe (wie Datenabstraktion, Klassenzusammenhalt und Objektkopplung) nicht nur umgangssprachlich zu interpretieren, sondern verwenden Sie diese Begriffe so wie in der Vorlesung und im Skriptum beschrieben. Als Hilfestellung sind in jeder Aufgabenstellung die Teile des Skriptums genannt, in denen die relevantesten Themen behandelt werden - diesmal ein ganzes Kapitel, manchmal (vor allem bei komplizierten Themen) nur wenige Seiten.

Versuchen Sie nicht, Teile der Aufgabenstellung durch Tricks oder Spitzfindigkeiten zu umgehen. Beispielsweise gibt es immer wieder Lösungen, in denen die Test-Klasse nur den String "Tests erfolgreich" ausgibt statt tatsächlich Tests durchzuführen. Solche Versuche werden durch händische Beurteilungen mit hoher Wahrscheinlichkeit erkannt. Spätere Aufgaben enthalten oft Schwierigkeiten, die mit Allgemeinwissen alleine oder über aufgabenbezogene Internet-Recherchen kaum zu lösen sind. Gerade in solchen Fällen ist davon abzuraten, die Schwierigkeiten durch Tricks zu umgehen. Hinweise zur richtigen Lösung lassen sich im Skriptum oder auf den Vorlesungsfolien finden.

### **Was im Hinblick auf die Abgabe zu beachten ist:**

Schreiben Sie Ihre Lösung in das am Beginn der Aufgabenstellung genannte Verzeichnis (*Gruppe/Aufgabe1* für die erste Aufgabe) auf der *g0*. Das Programm soll von diesem Verzeichnis aus durch den ebenfalls oben genannten Befehl (*java Test* für diese und einige weitere Aufgaben) aufrufbar sein; das heißt, die Methode *main* soll in der Klasse *Test* in der Datei *Test.java* (die Sie selbst schreiben) stehen. Schreiben Sie (abgesehen von geschachtelten Klassen) nie mehr als eine Klasse in jede Datei. Das Verzeichnis soll zum Zeitpunkt der Abgabe und der nachträglichen Abgabe alle *java*-Dateien enthalten, die Sie für Ihre Lösung benötigen. Bitte entfernen Sie alle *java*-Dateien, die nicht zur Abgabe gehören. Dateien mit anderen Endungen werden bei der Beurteilung

nicht berücksichtigt. Stellen Sie sicher, dass ein Aufruf von `javac *.java` (bzw. `javac *.java */*.java` und `javac *.java */*.java */*/*.java` und so weiter wenn Sie Pakete verwenden) im Abgabeverzeichnis keine Fehlermeldung liefert, da Lösungen mit Syntaxfehlern im Allgemeinen nicht beurteilt werden. Warnings sind bei den ersten Aufgaben zwar erlaubt, später aber teilweise nicht mehr. Versuchen Sie gleich von Anfang an, Warnings zu vermeiden. Verwenden Sie außer den Java-Standardbibliotheken keine vorgefertigten Programmteile, da diese bei der Beurteilung als nicht vorhanden angesehen werden. Vermeiden Sie jede Änderung der Verzeichnisstruktur der Abgabeverzeichnisse (kein Verschieben, Umbenennen, Löschen und Neuanlegen) um die automatische Abgabe nicht zu behindern.

### **Wie Account und Gruppen-Verzeichnis zu verwenden sind:**

Bei der Anmeldung zur Lehrveranstaltung haben Sie (jeder Teilnehmer) einen eigenen Account auf dem Übungsrechner *g0* bekommen. In diesem Account können Sie unabhängig von anderen Übungsteilnehmern (auch unabhängig von anderen Mitgliedern Ihrer Gruppe) arbeiten. Das Verzeichnis *Gruppe* gehört jedoch allen Gruppenmitgliedern gemeinsam. Legen Sie alle Daten, die von den Gruppenmitgliedern gemeinsam (und nicht nur von Ihnen persönlich) verwendet werden, in diesem Verzeichnis ab. Das gilt insbesondere für Lösungen von Aufgaben. Es kann pro Gruppe und Aufgabe nur eine gemeinsame Lösung geben, auf die sich die Gruppenmitglieder einigen müssen. Es bleibt der Gruppe überlassen, wie die Arbeit aufgeteilt wird, solange die Aufteilung einigermaßen gerecht ist (wird beim Abgabegespräch überprüft).

Bitte achten Sie darauf, dass alle Gruppenmitglieder Lese- und Schreibrechte auf Dateien im Verzeichnis *Gruppe* haben. Fehlende Rechte entstehen oft durch Kopieren einer Lösung vom eigenen Rechner in ein Abgabeverzeichnis. Der Besitzer von *datei* kann durch `chmod g+rw datei` (oder `chmod g+rw *`) allen Gruppenmitgliedern solche Zugriffe auf *datei* (bzw. alle seine Dateien im aktuellen Verzeichnis) geben. Manchmal führen fehlende Rechte dazu, dass letzte notwendige Änderungen vor der Abgabe nur mehr sehr schwer möglich sind, wenn der Besitzer einer Datei gerade nicht greifbar ist.

Generell sollten Sie Änderungen der Struktur von Abgabeverzeichnissen vermeiden, um die automatisierte Abgabe nicht zu behindern. Sie können aber in *Gruppe* (jedoch nicht in *Gruppe/Aufgabe1*, *Gruppe/Aufgabe2*, etc.) beliebig weitere Verzeichnisse anlegen und dort beispielsweise Backups oder andere Dateien, die nicht zur Abgabe gehören, ablegen.

### **Was Ihr Tutor von Ihnen wissen möchte:**

Wahrscheinlich hat Ihnen Ihr Tutor bereits eine Mail geschrieben, in der er sich vorstellt und um Informationen über Sie bittet. Geben Sie ihm diese Information möglichst bald, damit der Tutor die für Sie am besten geeignete Form der

Betreuung wählen kann. Unabhängig von der Form der Betreuung kann natürlich jedes Gruppenmitglied jederzeit konkrete Fragen an den Tutor richten. Scheuen Sie sich bitte nicht, sich auch mit organisatorischen oder gruppeninternen Problemen, die Sie möglicherweise nicht selbst lösen können, an den Tutor zu wenden. Je früher Tutoren von Problemen erfahren, um so einfacher sind sie zu lösen.