

## 2. Übungsaufgabe

### Themen:

Aufwandsabschätzung, Programmiereffizienz, Untertypbeziehungen und dynamisches Binden

### Termine:

Ausgabe: 15.10.2014

Abgabe: 22.10.2014, 12:00 Uhr

### Abgabeverzeichnis:

Aufgabe2+3

### Programmaufruf:

java Test

### Grundlage:

[Skriptum](#), Schwerpunkt auf Abschnitt 2.1

## Aufgabe

### Welche Aufgabe zu lösen ist:

Erweitern Sie Ihre Lösung der [ersten Aufgabe](#) so, dass das Surfstoresystem in der Praxis verwendbar wird. Von Anwendern des Systems werden unten aufgezählte Erweiterungen für sinnvoll oder notwendig erachtet. Diese Aufzählung ist einerseits noch unvollständig, andererseits aber bereits so umfangreich, dass in der vorgegebenen Zeit bei weitem nicht alles umsetzbar ist. Ihre Aufgabe besteht darin, eine vernünftige Teilmenge der aufgezählten Punkte zu wählen und in Programmcode umzusetzen. Ein möglichst großer Teil der Ihrer Meinung nach wichtigsten Funktionalität soll dadurch abgedeckt werden.

- Einkauf und Verkauf von neuen und gebrauchten Artikeln ist möglich.

Defekte Artikel können ausgemustert werden.

- Eine Lagerverwaltung der Verkaufs- und Verleihartikeln inklusive statistischen Auswertungen wird benötigt.
- Ausgaben für den Einkauf von Artikeln und Einnahmen für Verkauf und Verleih von Artikeln werden in einer Buchhaltung verwaltet. Die Geldbeträge werden sowohl auf ein Konto überwiesen als auch bar in einer Kasse ein- und ausgezahlt. Der Kassastand kann abgefragt werden.
- Rechnungen und Bestellungen können erstellt werden.
- Die Verleihdauer kann Stunden und Tage sein. Für längere Verleihdauern gibt es Rabatte.
- Den selben Artikel gibt es in unterschiedlichen Größen.
- Unterschiedliche Statistiken zum Verkauf und zum Verleih werden erstellt.
- Unterschiedliche Statistiken zum Umsatz und diverse Kassaberichte werden erstellt.
- Umsatzsteuer wird berücksichtigt und monatliche Umsatzsatzsteuermeldungen werden erstellt.
- Ein Bonusprogramm mit Rabatten zur Kundenbindung wird unterstützt, wenn ein Mindestjahresumsatz erreicht wird.
- Es wird noch eine Surfschule angeschlossen. Kurse, Termine, Lehrer, Schüler und die Zuordnungen untereinander werden verwaltet.
- Daten gehen bei Programmbeendigung nicht verloren, und ein Backup-System verhindert Datenverlust bei einem Systemausfall.

Schreiben Sie in Java wieder den Kern eines solchen Programms (ohne Benutzerschnittstelle) sowie ein Testprogramm. Der Kern soll die von Ihnen gewählte Funktionalität aufweisen, aber keinerlei Eingabe von der Tastatur oder Ausgabe auf den Bildschirm machen.

Das Testprogramm soll mittels *java Test* von *Aufgabe2+3* aus aufrufbar sein und die Funktionalität überprüfen. Die Tests sollen selbständig ohne Benutzerinteraktion ablaufen, sodass Aufrufer keine Testfälle auswählen oder Testdaten eintippen müssen. Testergebnisse sollen (in nachvollziehbarer und verständlicher Form) am Bildschirm ausgegeben werden.

Neben dem Testprogramm soll die Klasse *Test.java* als Kommentar eine kurze, aber verständliche Beschreibung der Aufteilung der Arbeiten auf die einzelnen Gruppenmitglieder enthalten - wer hat was gemacht.

### **Wie die Aufgabe zu lösen ist:**

Die oben aufgezählten gewünschten Eigenschaften sind nur als Anhaltspunkte gedacht. Sie können auch andere sinnvoll erscheinende Eigenschaften einbauen und einzelne Punkte nur unvollständig oder anders als angedeutet lösen.

Eine der größten Schwierigkeiten dieser Aufgabe besteht in der richtigen Abschätzung des Umfangs der Arbeiten, die Sie bis zum Abgabetermin fertigstellen können. Planen Sie entsprechend Ihren Vorkenntnissen und

Fähigkeiten möglichst viel ein, das Sie in der vorgesehenen Zeit auch zum Abschluss bringen können - das heißt, so viel Sie können, aber auf keinen Fall mehr als Sie können. Als groben Anhaltspunkt sollten Sie (jedes Gruppenmitglied) mindestens fünf bis sechs Stunden in die Lösung dieser Aufgabe fließen lassen. Versuchen Sie so effizient wie möglich zu arbeiten und sehr rasch zu einer brauchbaren Lösung zu kommen. Ignorieren Sie Details, die Ihnen als unwichtig erscheinen. Bedenken Sie, dass Sie alle Teile Ihrer Lösung durch Testfälle überprüfen sollen und planen Sie die Zeit für die Entwicklung der Testfälle und für die Fehlerbeseitigung ein.

Senden Sie möglichst bald ein grobes Konzept Ihrer geplanten Arbeiten - am besten mit einer Einteilung, welches Gruppenmitglied was machen soll - an Ihre(n) Tutor(in). Die Tutor(inn)en werden sich bemühen, rasch hilfreiche Rückmeldungen zu den Konzepten zu geben. Überschätzen Sie sich nicht. Erstellen Sie eher ein Konzept, von dem Sie sicher annehmen, dass Sie die Arbeiten zeitlich schaffen. Wenn das Konzept Ihrer Tutorin oder Ihrem Tutor nicht reicht, werden Sie um eine Abänderung gebeten.

Einer der Schwerpunkte dieser Aufgabe ist der Umgang mit Untertypbeziehungen. Planen Sie die Verwendung von Untertypbeziehungen zusammen mit dynamischem Binden ein.

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

Sie sollen möglichst große Freiheit bei der Lösung der Aufgabe haben und selbst die Verantwortung für alles übernehmen. Niemand schreibt Ihnen vor, wie die Aufgabenstellung genau zu verstehen ist.

Diese Aufgabe stellt hohe Anforderungen an jedes einzelne Gruppenmitglied sowie die Zusammenarbeit innerhalb der Gruppe - eine Nagelprobe für das Funktionieren der Gruppe und zum Aufdecken möglicher Schwachstellen.

Untertypbeziehungen sind ein schwieriges, aber für die objektorientierte Programmierung sehr wichtiges Thema. Nutzen Sie die Gelegenheit, bei der Lösung der Aufgabe Erfahrungen damit zu gewinnen. Fehler, die Sie dabei noch machen, wirken sich nicht auf Ihre Beurteilung aus. Sie bekommen von den Tutor(inn)en Rückmeldungen und bei Bedarf maßgeschneiderte Hilfe.