FAKULTÄT FÜR INFORMATIK
Faculty of Informatics

Objektorientierte Programmiertechniken
LVA 185.A01, VL 2.0, 2011 W

2. Übungsaufgabe

Themen:
Aufwandsabschätzung, Programmiereffizienz, Untertypbeziehungen und dynamisches Binden

Termine:

Ausgabe:	19.10.2011
1. Vorabgabe:	26.10.2011, 13:45 Uhr
2. Vorabgabe:	02.11.2011, 13:45 Uhr
reguläre Abgabe:	09.11.2011, 13:45 Uhr
nachträgliche Abgabe:	16.11.2011, 13:45 Uhr

Abgabeverzeichnis:
Gruppe/Aufgabe2

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 Anmeldesystem 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.

- Entfernen einer Lehrveranstaltung (mit allen Daten) aus der Liste der sichtbaren Lehrveranstaltungen. Dabei sollen die Daten jedoch nicht verlorengehen, sondern bei Bedarf wieder rekonstruiert werden können.
- Es soll möglich sein, Daten zur Lehrveranstaltung (insbesondere An- und Abmeldetermine) zu ändern oder die Lehrveranstaltung

gänzlich abzusagen, obwohl bereits Studierende dazu angemeldet sind. Diese Studierenden sind auf geeignete Weise über die Änderungen zu informieren.

- Regeln im Zusammenhang mit Anmeldungen sollen besser im System abgebildet werden: Anmeldungen zu bestimmten Lehrveranstaltungen setzen eine abgeschlossene STEOP voraus, andere wiederum ein abgeschlossenes STEG. Ganz allgemein soll die Anmeldung zu bestimmten Lehrveranstaltungen von der vorhergehenden Absolvierung anderer Lehrveranstaltungen abhängen können. Ausnahmen von der Regel müssen in Sonderfällen handhabbar sein, aber nicht jeder darf beliebige Ausnahmen zulassen.
- Man kann sich über das System nicht nur zu Lehrveranstaltungen, sondern auch zu Prüfungen, Abgabegesprächen, etc. anmelden.
- Im Zuge der Anmeldung sollen je nach Lehrveranstaltung automatisiert weitere Schritte gesetzt werden können, beispielsweise die Bildung von Gruppen, Zuweisung von TutorInnen, Vergabe von Accounts auf Übungsrechnern, Einrichtung von Diskussionsforen, sowie Verteilung von Skripten und individuellen Aufgabenstellungen.
- Im System werden alle Leistungen der Studierenden (erreichte Punkte bei Übungsaufgaben, Prüfungsantritte und dabei erzielte Beurteilungen, etc.) erfasst.
- Eine maximale Anzahl an Teilnehmern bei Lehrveranstaltungen, Prüfungen, etc. soll festlegbar sein. Um mehr Gerechtigkeit bei der Anmeldung zu überlaufenen Lehrveranstaltungen und Prüfungen gewährleisten zu können, sollen Informationen dazu bereits vor einem festgelegten frühesten Anmeldezeitpunkt gegeben werden.
- Lehrveranstaltungen können ganz unterschiedlich strukturiert sein – Vorlesungen, Übungen, etc. – mit einer (schriftlichen oder mündlichen) Prüfung am Ende oder beliebig vielen Beurteilungsschritten unterschiedlicher Art – ganz oder teilweise in Gruppen oder jeder für sich – und so weiter. Um die Festlegung der genauen Modalitäten zu erleichtern, werden Lehrveranstaltungen in Lehrveranstaltungsarten eingeteilt. Lehrveranstaltungsarten können unterschiedliche Aspekte betreffen – Art der Anmeldung, Art der Beurteilung, etc. und dementsprechend kann jede Lehrveranstaltung mehrere Lehrveranstaltungsarten haben. Auch für einzelne Übungsaufgaben, Abgabegespräche, Prüfungen und so weiter sind Arten festlegbar. Eine Lehrveranstaltungsart kann aus anderen (einfacheren) Lehrveranstanstaltungsarten und Arten ihrer Bestandteile (Übungsaufgaben, etc.) zusammengesetzt werden.
- Wegen der umfangreichen personenbezogenen Daten im System muss sichergestellt sein, dass nur jene Benutzer Zugang erhalten, die dazu berechtigt sind. Studierende dürfen nur Zugang zu den sie selbst betreffenden Daten erhalten, Tutoren nur zu den die eigenen Gruppen betreffenden Daten und Lehrveranstaltungsleiter nur zu den die eigenen Lehrveranstaltungen betreffenden Daten. Nur wenige Administratoren bekommen Zugang zu allen Daten. Einige Daten – beispielsweise die angebotenen

Lehrveranstaltungen und Prüfungen – sollen unbeschränkt sichtbar sein.
- Daten für die Zugangskontrolle müssen verwaltet werden. Als Grundlagen dienen Daten aus der Immatrikulation und Inskription von Studierenden, Dienstverträge von Lehrenden, sowie Genehmigungen zur Abhaltung von Lehrveranstaltungen durch die Universitätsverwaltung. Es muss sichergestellt sein, dass der Zugang zu den Daten erlischt, sobald die Grundlage dafür nicht mehr gegeben ist.
- Lehrende und Studierende sollen von jeder Änderung, die sie betrifft, informiert werden.
- Daten gehen bei Programmbeendigung nicht verloren, und ein Backup-System verhindert Datenverlust bei einem Systemausfall.

Entwickeln Sie nach wie vor nur den Kern eines entsprechenden Programms ohne Benutzerschnittstelle. Dieser soll keine Eingabe von der Tastatur verlangen oder Ausgabe auf den Bildschirm machen.

Erweitern Sie das Testprogramm (aufrufbar mittels *java Test* von *Gruppe/Aufgabe2* aus) um Überprüfungen der zusätzlichen Funktionalität. Wie in der ersten Aufgabe sollen alle Tests selbständig ohne Benutzereingaben ablaufen und Testergebnisse in nachvollziehbarer und verständlicher Form am Bildschirm ausgeben.

Wie die Aufgabe zu lösen ist:
Die oben aufgezählten möglichen Erweiterungen sind nur als Anhaltspunkte gedacht. Sie können auch andere sinnvoll erscheinende Erweiterungen machen und einzelne Punkte nur unvollständig oder anders als angedeutet lösen.

Eine der größten Schwierigkeiten dieser Aufgabe besteht in der richtigen Abschätzung des Umfangs der Erweiterungen, die Sie bis zum regulären Abgabetermin machen können. Planen Sie entsprechend Ihren Vorkenntnissen und Fähigkeiten möglichst viele Erweiterungen ein, die 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) etwa fünf bis sechs Stunden pro Woche in die Lösung dieser Aufgabe fließen lassen – also insgesamt etwa 50 Stunden (ca. 3 * 3 * 5,5 bei drei Gruppenmitgliedern). Versuchen Sie, 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. Es wird im Normalfall keine Gelegenheit geben, Teile der Arbeit, die Sie in der vorgegebenen Zeit nicht zu Ende bringen, nachzuholen. In der Zeit zwischen regulärem und nachträglichem Abgabetermin werden Sie bereits mit der nächsten Aufgabe beschäfftigt sein.

Senden Sie möglichst bald ein grobes Konzept Ihrer geplanten Erweiterungen (am besten mit einer Einteilung, welches Gruppenmitglied was machen soll) an Ihren Tutor. Die Tutoren 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 problemlos schaffen. Wenn das Konzept Ihrem Tutor nicht reicht, werden Sie ohnehin rasch um eine Abänderung gebeten werden.

Vor der regulären Abgabe sind zwei Termine für Vorabgaben vorgesehen. Bitte stellen Sie Ihre unvollständigen Lösungen vor den Vorabgabeterminen in das Abgabeverzeichnis auf der g0. Damit erhalten die Tutoren Informationen darüber, wie weit Sie schon gekommen sind, und erkennen vielleicht auch Probleme in Ihrem Lösungsansatz. Wenn etwas nicht passt, werden Sie gleich darüber informiert werden und haben noch genug Zeit für Korrekturen. Die Vorabgaben sind freiwillig. Sie müssen sie also nicht machen, aber beide Vorabgaben werden dringend empfohlen.

Einer der Schwerpunkte dieser Aufgabe ist der Umgang mit Untertypbeziehungen zusammen mit dynamischem Binden. Planen Sie Ihre Erweiterungen so, dass Untertypbeziehungen verwendet werden. Setzen Sie gezielt dynamisches Binden ein.

Beachten Sie die allgemeinen Informationen zur Laborübung aus der ersten Aufgabe und stellen Sie Ihre Lösung in das Verzeichnis *Gruppe/Aufgabe2*. Von dort aus soll das Testprogramm durch *Java Test* aufrufbar sein. Wegen des Umfangs dieser Aufgabe ist es gestattet, Unterverzeichnisse von *Gruppe/Aufgabe2* anzulegen und mit Paketen zu arbeiten. Sie müssen aber keine Pakete verwenden, wenn Sie darin keine Vorteile sehen.

Warum die Aufgabe diese Form hat:
Sie sollen möglichst große Freiheit bei der Lösung der Aufgabe haben und selbst die Verantwortung für alles übernehmen, was mit der Entwicklung der Software zu tun hat. Es gibt niemanden, der Ihnen vorschreibt, wie die Aufgabenstellung genau zu verstehen ist. Dadurch sollen Sie eine Ahnung davon bekommen, was es bedeutet, selbst für ein Softwareprojekt verantwortlich zu sein.

Diese Aufgabe ist im Vergleich zu allen anderen Aufgaben die bei weitem umfangreichste und stellt damit hohe Anforderungen an jedes einzelne Gruppenmitglied sowie die Zusammenarbeit innerhalb der Gruppe – eine Nagelprobe für das Funktionieren der Gruppe und zum Aufdecken möglicher Schwachstellen in der Organisation der Gruppe.

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

Anfang | HTML 4.01 | letzte Änderung: 2011-10-19 (Puntigam)