Objektorientierte Programmiertechniken
LVA 185.A01, VU, 3 Ects, 2012/2013 W

2. Übungsaufgabe

Themen:

Aufwandsabschätzung, Programmiereffizienz, Untertypbeziehungen und dynamisches Binden

Termine:

Ausgabe:	17.10.2012
reguläre Abgabe:	24.10.2012, 12:00 Uhr
nachträgliche Abgabe:	31.10.2012, 12:00 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 Verwaltungssystem in der Praxis verwendbar wird. Von Anwendern 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 nicht alles umsetzbar ist. Ihre Aufgabe besteht darin, eine vernünftige Teilmenge der aufgezählten sowie von Ihnen selbst hinzugefügten Punkte zu wählen und in Programmcode umzusetzen. Ein möglichst großer Teil der wichtigsten Funktionalität soll dadurch abgedeckt werden.

- Entfernen oder Ändern einer Probe oder eines Auftritts ist vor allem zur Korrektur falscher Daten notwendig. Dabei sollen die entfernten oder geänderten Daten nicht verlorengehen, sondern bei Bedarf wieder rekonstruiert werden können.
- Es soll möglich sein, Proben und Auftritte anzusetzen, zu verschieben und abzusagen. Darüber sollen die Mitglieder der Musikgruppe automatisch informiert werden.
- Die Mitglieder der Musikgruppe sollen in die Planung der Proben und Auftritte eingebunden werden, indem Sie entsprechenden Terminvorschlägen zustimmen oder diese ablehnen. Die Entscheidungen sollen mit kurzen Begründungen versehen werden können.
- Nachdem Auftritte und vor allem Proben häufig am selben Ort stattfinden, sollte es eine eigene Verwaltung dieser Orte geben. Die wichtigste Infrastruktur dieser Orte sollte leicht auffindbar beschrieben sein. Man soll geeignete Orte auch anhand bestimmter Infrastruktur wählen können.
- Neben Raummieten und Gagen fallen viele weitere Kosten und gelegentlich auch Einnahmen an. Diese sollen direkt im System verwaltet werden. Die Zuordnung zwischen Auftritten bzw. Proben und Einnahmen bzw. Kosten soll weitestgehend erhalten bleiben, aber auch Einnahmen und Kosten, die nicht eindeutig zuzuordnen sind, sollen abgebildet werden.
- Für das Aufsummieren der Einnahmen und Kosten in einem bestimmten Zeitraum sollen Filter einsetzbar sein, die bestimmen, welche Arten von Einnahmen und Kosten zu berücksichtigen sind.
- Die Musikgruppe hat neben permanenten Mitgliedern auch Ersatzmitglieder, die nur im Bedarfsfall zum Einsatz kommen. Pro Probe und Auftritt kann sich die Zusammensetzung unterscheiden und ist im System festzuhalten. Es kann auch zu Verschiebungen kommen, wobei ein permanentes Mitglied zu einem Ersatzmitglied wird und umgekehrt. Ersatzmitglieder, die nicht an einer Mindestzahl an Proben pro Zeiteinheit teilnehmen, werden für Auftritte gesperrt.
- Das Repertoire an Musikstücken hängt von der aktuellen Besetzung bei einem Auftritt und der Teilnahme an Proben ab. Es gibt also nicht nur ein gemeinsames Repertoire zu jedem Zeitpunkt, sondern jedes Gruppenmitglied hat sein persönliches Repertoire (zu jedem Zeitpunkt), und das Gesamtrepertoire ergibt sich aus der jeweiligen Zusammensetzung.
- Zu jedem Musikstück kann es mehrere Varianten geben. Beim Auflisten können Varianten entweder getrennt voneinander oder gemeinsam betrachtet werden. Bei gemeinsamer Betrachtung kann ein Musikstück unterschiedliche Längen aufweisen.
- Zu Planungszwecken soll im System nicht nur die Vergangenheit darstellbar sein, sondern auch die erwartete Zukunft. Alle Daten können auch geplante Vorgänge widerspiegeln. Um Verwechslungen zu vermeiden, sollen erwartete zukunftsbezogene Daten klar von bestätigten Daten aus der Vergangenheit unterschieden werden. Erwartete Daten sollen zu gegebener Zeit als bereits zugetroffen bestätigt und dadurch in bereits vergangene Daten übergeführt werden können. Natürlich sollen Daten

auch entfernt werden können, wenn Erwartungen nicht eingetroffen sind. Eine Suchfunktion nach zukunftsbezogenen Daten, die aufgrund ihres Datums schon bestätigt oder entfernt sein sollten, wird benötigt.
- Es muss sichergestellt sein, dass nur jene Benutzer Zugang zum System erhalten, die dazu berechtigt sind. Gruppenmitglieder sollen Zugang zu den sie selbst betreffenden Daten erhalten, die Leitung zu den Daten aller Gruppenmitglieder, die Verwaltung vor allem zu den Einnahmen und Kosten, und so weiter. Insgesamt müssen alle Daten gewartet werden können, andererseits soll niemand auf Daten zugreifen dürfen, die ihn oder sie nicht betreffen. Nur wenige Daten sind für die Allgemeinheit bestimmt.
- Daten für die Zugangskontrolle müssen verwaltet werden. Dabei ist auch sicherzustellen, dass der Zugang zu den Daten nicht mehr möglich ist, sobald die Grundlage dafür wegfällt, etwa weil ein Mitglied ausscheidet.
- Daten gehen bei Programmbeendigung nicht verloren, und ein Backup-System verhindert Datenverlust bei einem Systemausfall.
- Wenn es sich bewährt, soll das System nicht nur zur Verwaltung einer Musikgruppe, sondern zum professionellen Management vieler Gruppen eingesetzt werden. Dafür sind Vorkehrungen zu treffen.

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 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. 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 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 Ihrem Tutor oder Ihrer Tutorin nicht reicht, werden Sie um eine Abänderung gebeten.

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. Auch wenn die Aufgabe nicht dahin getrimmt ist, dass sich Untertypbeziehungen von alleine ergeben, lassen sich sicher gute Gelegenheiten dafür finden.

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. Es gibt niemanden, der Ihnen vorschreibt, 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.