

Übung Informatik IIb

(Wintersemester 2010/11)

Oliver Jack
Fachhochschule Jena
Fachbereich Elektrotechnik und Informationstechnik

Oktober 2010

1 Organisatorisches

Dieses Dokument beschreibt den Inhalt und das Verfahren der Alternativen Prüfungsleistung für die Lehrveranstaltung *Informatik IIb* im Wintersemester 2010/11.

- Bilden Sie Projektgruppen mit einer Sollstärke von vier Studierenden gemäß Seiteinteilung.
- Wählen Sie pro Projektgruppe eines der in Abschnitt 4 folgenden drei Software-Entwicklungsprojekte aus.
- Die Projektgruppen und die Wahl der Software-Entwicklungsprojekte wird in der zweiten Übung festgelegt.
- Lösen Sie die gewählte Aufgabe in Gruppenarbeit bis zum vorletzten Praktikumstermin.
- *Präsentieren Sie Ihre Lösung.* Stellen Sie in der Präsentation Ihren Entwurf, die Realisierung, die qualitätssichernden Maßnahmen vor und demonstrieren Sie Ihr Softwareprodukt. Jedes Projektgruppenmitglied präsentiert einen Teil der Lösung. Für die Präsentation stehen pro Gruppe *zehn Minuten* zur Verfügung. Anschließend stehen *fünf Minuten für Fragen* aus dem Auditorium, einschließlich Dozent, zur Verfügung. Jedes Projektgruppenmitglied muss einen angemessenen Anteil an der Präsentation haben, das heißt, jedes Projektgruppenmitglied präsentiert ca. 2 1/2 Minuten.
- Die Präsentationen finden am letzten Praktikumstermin statt. Die Praktikumstermine finden Sie unter <http://stundenplanung.fh-jena.de/>.

- Geben Sie pro Projektgruppe *spätestens zum Präsentationstermin* eine *schriftliche Ausarbeitung* Ihrer Projektarbeit ab. Die Abgabe erfolgt *ausschließlich per E-Mail* an oliver.jack@fh-jena.de unter Angabe des Betreffs „Info Iib APL WS 2010/11“. Im Text der E-Mail geben Sie bitte Ihre Praktikumsgruppe, Ihre Projektgruppennummer und alle Projektgruppenmitglieder mit Namen und Matrikelnummer an. Bitte geben Sie nur eine Datei im ZIP- oder RAR-Format ab. Die schriftliche Ausarbeitung muss folgende Teile enthalten:
 1. Präsentationsfolien im PDF-Format,
 2. Beschreibung des Entwurfs und der Arbeitspakete im PDF-Format,
 3. Zuordnung der Arbeitspakete zu den einzelnen Projektgruppenmitgliedern im Textformat,
 4. kommentierten Quell-Code im Textformat (ASCII) oder annotierten Quell-Code im doxygen-html-Format,
 5. Testfälle mit Klassifikation und Testauswahlkriterien im Tabellendokument (ODS oder Microsoft Excel),
 6. Testprotokoll im Tabellendokument (ODS oder Microsoft Excel),
- Weitere Informationen und Hinweise zur Lehrveranstaltung und der Alternativen Prüfungsleistung finden Sie auf der Website <http://www.et.fh-jena.de/jack/> bzw. <http://www.fh-jena.de/~jack/>.

2 Bewertungskriterien

Für die Bewertung sind maßgeblich

- Verständlichkeit von Softwareentwurf (UML-Klassendiagramme) und Implementierung (Struktur, Übersichtlichkeit und Kommentierung des Quellcodes),
- Qualität der Softwaretests (vollständige Testfalllisten, Testauswahlkriterien, vollständige Testprotokolle),
- Funktionstüchtigkeit und Validität der Implementierung,
- Qualität der Präsentation und Vorführung (Vortragsstil, Strukturierung, Konzentration auf das Wesentliche).

Hinweise zu den Testfalllisten und Testprotokollen:

Eine *Testfallliste* besteht aus einer tabellarischen Angabe von

1. Testeingabe,
2. Zustand der zu testenden Software (z. B. Einstellung "Datenbank www.freedb.org" im Menü "Verbindungen" gewählt),
3. erwartete Ausgabe und erwarteter Zustand der Software (z. B. Anzeige der Titelliste im Fenster „Audiofiles“).

Ein *Testprotokoll* besteht aus einer tabellarischen Angabe von

1. Testfall gemäß Testfallliste,
2. Version der getesteten Software
3. Datum und Uhrzeit der Testdurchführung,
4. Ergebnis des Tests („bestanden“, „nicht bestanden“; bei „nicht bestanden“ Angabe der aktuellen Ausgabe bzw. des aktuellen Zustands der getesteten Software.

3 Hinweise zur Bearbeitung

- Beginnen Sie bei der Bearbeitung der Aufgabe mit der Modellierung der Datenstrukturen (Klassen), nutzen Sie die Konzepte der objektorientierten Programmierung (Datenkapselung, Polymorphismus, Vererbung etc.) und der Standard-Java-Pakete. Setzen Sie diese Konzepte sach- und fachgerecht ein.
- Teilen Sie die Bearbeitung der Aufgabe auf die einzelnen Projektgruppenmitglieder auf, nachdem Sie die Klassenstruktur und die Schnittstellen der Klassen (Attribute und Methoden) festgelegt haben, so dass die Implementierung der einzelnen Klassen arbeitsteilig erfolgen kann.

4 Semesterprojekte

Wählen Sie je Projektgruppe eine der nachfolgenden Aufgaben aus. Bearbeiten Sie die Aufgabe unter Verwendung der Programmiersprache Java. Beachten Sie die in Abschnitt 2 festgelegten Rahmenbedingungen für die alternative Prüfungsleistung.

4.1 Bib_TE_X-Datenbank

Entwickeln Sie eine Software zur Bearbeitung und Verwaltung von Bib_TE_X-Datenbanken. Bib_TE_X ist eine Software, mit deren Hilfe Literaturdatenbanken aufgebaut und Literaturangaben in L^AT_EX-Dokumente eingebunden werden können. Das Bib_TE_X-Datenbankformat ist eine strukturierte Textdatei der Form

```
@article{mrx05,  
author = {Mr. X},  
title = {Something Great},  
publisher = {nobody},  
year = 2005,  
}
```

Das Format lässt viele verschiedene Einträge zu, die mit dem Dokumenttyp beginnen, z. B. `@article` für einen Zeitschriftenartikel, `mrx05` als Eintragungsschlüssel, und viele Eintragungsfelder besitzen, z. B. `author` für den Autor des Artikels. Manche dieser Felder sind optional, müssen also nicht angegeben sein, andere sind obligatorisch, müssen also vorhanden sein. Welche Felder obligatorisch und welche optional sind, hängt vom Eintragungstyp ab. Bei der Angabe der Feldnamen, z. B. `author`, wird nicht zwischen Groß- und Kleinschreibung unterschieden. Eine Beispiel- und Testdatei mit Bib_TE_X-Einträgen finden Sie auf der Website der Informatik Iib-Übung:

<http://www.fh-jena.de/~jack/Lehre/InfoIib/Uebung/xampl.bib>,
<http://www.fh-jena.de/~jack/Lehre/InfoIib/Uebung/test.bib>).

Weitere Informationen zu Bib_TE_X finden sie z. B. hier: <http://www.bibtex.org/>

Die von Ihnen zu entwickelnde Software soll eine grafische Benutzeroberfläche besitzen und soll

- mehrere Bib_TE_X-Dateien laden können,
- die darin enthaltenen Einträge auf Korrektheit und Konsistenz prüfen, ggf. eine Bildschirm-Ausgabe erzeugen, die die unkorrekten bzw. inkonsistenten Einträge anzeigt,
- neue Literatureinträge anlegen und bestehende bearbeiten als auch löschen können, sowohl durch Tastatureingabe als auch durch Eingabe über den Copy-Paste-Puffer, bei letzterem ist der Inhalt des Paste-Puffers ein Text wie oben im Beispiel

```
@article{mrx05,,
```

- Freitextsuche in den Feldern der Eintragstypen ermöglichen, d. h. die gefundenen Einträge werden mit Ihrem Schlüssel in einer Liste angezeigt, in einem anderen Fenster können die gefundenen Einträge im Detail betrachtet und bearbeitet werden,
- eine Auswahl beliebiger Einträge als neue BibTeX-Datenbankdatei schreiben können,
- ein zusätzliches optionales Feld für jeden Eintragstyp mit dem Bezeichner **abstract** beinhalten,
- ein weiteres zusätzliches optionales Feld für jeden Eintragstyp mit der Bezeichnung **link** beinhalten, das einen Hyperlink zu einem Dokument (das wird sinnvollerweise das Dokument, das im Eintrag beschrieben ist, sein) enthält; bei einem Mausklick auf den Link soll dieser Link mit dem auf den Zielsystem verbundenen Programm geöffnet werden, z. B. bei einem PDF-Dokument mit dem Adobe-Reader.

4.2 Audioplayer

Entwickeln Sie eine Software, die einen Audioplayer realisiert. Die Software soll eine grafische Benutzeroberfläche besitzen und folgende Funktionen beinhalten:

1. Auswahl von mehreren Audiodateien über eine Dateiauswahlbox und über drag-and-drop aus einem Windows-Explorer-Fenster, bzw. entsprechenden Linux-Konstrukten, mindestens Microsoft Wave-Audioformate sollen verarbeitbar sein,
2. Anordnung der Auswahl zu einer Abspielreihenfolge, Funktionen: Sortieren, Zufallsreihenfolge, Anordnung über Positionen in der Liste per Maus-drag- and-drop,
3. Löschen von Elementen der Auswahl,
4. Abspielen der gewählten Audiodateien in der vorgesehenen Reihenfolge, Funktionen Play, Pause, Stop, Lautstärke, Anzeige der aktuellen Abspielposition über Progressbar und Minuten:Sekunden,
5. Verknüpfung der Audiodateien mit Metadaten gemäß IDv2-Tags und Laden sowie Speichern von Playlists, z. B. cue-Sheet oder m3u.
6. Laden von Metadaten aus der freedb (<http://www.freedb.org>)
7. Sie können zur Realisierung der Abspielfunktionen das Java Media Framework verwenden (<http://www.oracle.com/technetwork/java/javase/index-142695.html>).

4.3 Rhythmomachia

Entwickeln Sie eine Software, die das mittelalterliche Brettspiel *Rhythmomachia* realisiert. Die Software soll in einer grafischen Benutzeroberfläche folgende Funktionen beinhalten:

1. Anzeige des Spielbretts und der Spielsteine,

2. Erstellen der Spielanfangsaufstellung,
3. Realisierung der gültigen Spielzüge per Maus-drag-and-drop, Spielstein bewegen, Schlagen, Verweigern ungültiger Spielzüge,
4. Schaltbarer Modus, in dem zu einem per Maus gewählten ziehenden Spielstein die gültigen Felder eines Zuges angezeigt werden,
5. Anzeige und Beendigung des Spiels, wenn eine Gewinnsituation erreicht ist,
6. Rückgängigmachen des letzten Spielzugs

Die Spielregeln von Rhythmomachia finden Sie auf der Website der Informatik Iib-Übung (<http://www.fh-jena.de/~jack/Lehre/InfoIib/Uebung/Rhythmomachia.PDF>). Sie sind dem Buch „Rhythmomachia — Ein uraltes Zahlenspiel neu entdeckt“ von Detlef Illmer und Nora Gädeke, Elisabeth Henge, Helene Pfeiffer und Monika Spicker-Beck, Hugendubel Verlag, München, 1987, entnommen.