Richtlinien und Hinweise für die MW-Übungsaufgaben

- Jeder Student bekommt ein Projektverzeichnis unter /proj/i4mw/*<loginname>*.
- In diesem Projektverzeichnis muss für jede Übungsaufgabe ein Unterverzeichnis angelegt werden, z.B. für die fünfte Aufgabe mit:
  `mkdir /proj/i4mw/`*<loginname>*`/aufgabe5`
- Die Aufgaben müssen mit dem *abgabe*-Programm abgegeben werden. Um z.B. die fünfte Aufgabe abzugeben:
  `/proj/i4mw/pub/abgabe aufgabe5`
- Die Lösungen müssen eigenständig erstellt worden sein. Verstöße werden geahndet.
- Die *Java Coding Guidelines* sollen befolgt werden:
  - Klassennamen beginnen mit einem *Großbuchstaben*.
  - Methodennamen beginnen mit einem *Kleinbuchstaben*.
  - Variablennamen beginnen mit einem *Kleinbuchstaben*.
  - Konstante (final static) Variablen bestehen nur aus *Großbuchstaben*.
  - Variablenamen sind ausdrucksfähig und beschreiben deren Zweck.
- Wenn eine bestimmte Programmstruktur in der Aufgabenstellung verlangt wird, muss die Lösung diese einhalten. Die Lösung muss
  - Namen von Klassen und Packages
  - Sichtbarkeit von Variablen und Methoden
  - Namen, Parametertypen und Rückgabewerte von Methoden

  wie in der Aufgabenstellung beschrieben verwenden.
- Es werden Punkte abgezogen, falls
  - verlangte Funktionalität nicht implementiert wird
  - nicht verlangte Funktionalität implementiert wird
  - das Programm nicht kompiliert.
- Der Code muss lesbar und nachvollziehbar sein. Falls komplizierte Teile vorkommen, sollten diese mit Kommentaren erläutert werden.
- Bei Problemen mit der Aufgabenstellung könnt ihr euch jederzeit an uns wenden:
  {felser, weissel}@informatik.uni-erlangen.de

© 2003 Universität Erlangen Nürnberg, Institut für Informatik 4

Übungsaufgabe #1: Library - Erste Schritte

29.10.2003

In dieser Aufgabe wird das Grundgerüst für eine Bibliotheksverwaltung erstellt, das in den folgenden Aufgaben ausgebaut werden soll. Es sollen Klassen für Medienobjekte (Buch, CD, Zeitschrift) erstellt, sowie eine einfache Datenbank implementiert werden, welche diese Objekte verwaltet.

a) Erstelle ein Interface `Item`, das die gemeinsamen Methoden aller Medienobjekte definiert. Im Hinblick auf spätere Aufgaben sind mindestens folgende Methoden vorzusehen:

   `String getTitle()` und `void setTitle(String title)`

   Zum Setzen und Abfragen des Titels.

   `void borrow()` und `void giveback()`

   Objekt ausleihen bzw. zurückgeben. Falls es bereits verliehen ist, soll eine `AlreadyBorrowedException` geworfen werden.

   `void incLendingcount()` und `int getLendingcount()`

   Zur Verwaltung eines Ausleih-Zählers.

b) Erstelle drei Klassen `Book`, `Journal` und `CD`, die alle das Interface `Item` implementieren. Ein Konstruktor soll die Initialisierung mit einem Titel erlauben.

c) Schreibe die Datenbank-Klasse `LibraryDBImpl` zur Verwaltung der Item-Objekte. Dazu ist ein Interface `LibraryDB` mit mindestens folgenden Methoden zu implementieren:

   `int register(Item item)`

   Fügt das Item-Objekt der Datenbank hinzu und gibt eine eindeutige ID zurück.

   `Item get(int id)`

   Liefert eine *Kopie* des Item-Objekts mit der entsprechenden ID zum lesenden Zugriff zurück.

   `Item lock(int id)`

   Wie `get()`, jedoch für schreibenden Zugriff. Für einen erneuten Aufruf von `lock()` muss das Objekt mit `unlock()` wieder freigegeben werden.

   `void unlock(Item item)`

   Gibt das Objekt wieder frei.

   Eine mögliche Implementierung könnte darin bestehen, Item-Objekte in zwei verschiedenen Hashtabellen zu speichern, eine für gesperrte und eine für zugreifbare Objekte.

   Im Fehlerfall sollen geeignete Exceptions generiert werden (z.B. `AlreadyExistsException`, `NotFoundException` und `AlreadyLockedException`).

d) Schreibe eine Klasse `LibraryFrontend`, die als Schnittstelle zwischen dem Benutzer und der Datenbank dient und mindestens folgende Methoden aufweist:

   `void registerItem(String classname, String title)`

   Fügt ein neues Objekt der Datenbank hinzu. Durch `classname` wird der Typ des Objekts angegeben (`Book`, `CD` oder `Journal`).

   `void borrowItem(String title)`

   Leiht das Objekt aus (Ausleih-Zähler erhöhen!).

   `int returnItem(String title)`

   Das Objekt wird wieder zurückgegeben. Der Rückgabewert entspricht dem Ausleih-Zähler.

   Fehler sind durch geeignete Exceptions auszudrücken (z.B. `NotFoundException`, `AlreadyExistsException` und `AlreadyBorrowedException`).

   Die Kommunikation mit dem Benutzer soll über eine Eingabemaske ähnlich einer Shell realisiert werden. Zum Registrieren eines neuen Buches könnte z.B. folgender Aufruf dienen:

   ```
   > register Book "Operating Systems"
   ```

Bearbeitung: bis zum 06.11.2003/18:00 Uhr
Alle Dateien sollen im Verzeichnis `/proj/i4mw/`*`loginname`*`/aufgabe1/` abgelegt werden.

© 2003 Universität Erlangen Nürnberg, Institut für Informatik 4