

Die Lösungen zu diesem Blatt müssen zum zweiten Praktikum 11.10/12.10 vorliegen.

Nutzen Sie die unter C:\kleukersSEU von Ihnen installierte Software, siehe auch <http://kleuker.iui.hs-osnabrueck.de/kleukersSEU/index.html>.

Aufgabe 1 (4 Punkte)

Die Stadt Wiesbaden bietet in den Schulferien Ferienaktionen für Kinder und Jugendliche an. Um den enormen Andrang auf die Plätze in diesen Aktionen zu koordinieren, wurde 2006 eine Web-Lösung genutzt. Dabei wurden bei der Nutzung alle Kurse mit noch freien Plätzen angezeigt. Bei Aktionen, die noch Plätze frei hatten, konnte man mit einem Haken diese Aktionen markieren, in einer folgenden Seite seine Kontaktdaten eingeben und dann die Bestellung per Knopfdruck bestätigen. Das Resultat war, dass für einige Aktionen wesentlich mehr Plätze vergeben wurden, als vorhanden waren.

- Stellen Sie aufgrund Ihrer Datenbankkenntnisse eine begründete Vermutung an, warum das Problem auftrat.
- Gehen Sie alle aus der Vorlesung bekannten neun Aufgaben eines DBMS durch und überlegen Sie möglichst für jeden Punkt jeweils einen Fehler, den es in dem System noch geben könnte, wenn dieser Punkt nicht ordentlich umgesetzt würde.
- Beschreiben Sie ein Verfahren, so dass die Platzverteilung funktioniert. Welche Probleme könnte Ihr Verfahren haben (technisch bzw. aus Nutzersicht)?

Aufgabe 2 (4 Punkte)

Laden Sie von der Webseite der Veranstaltung das Projekt dbAufgabeDBSchreiben, mit dem Objekte einer Klasse Student verwaltet werden sollen.

- Schauen Sie sich den vorhandenen Code an. Nutzen Sie <http://kleuker.iui.hs-osnabrueck.de/querschnittlich/SEU.pdf> zur Einarbeitung in Eclipse.
- Schreiben Sie eine Implementierung des Interfaces `persist.VerwaltungInterface` und ergänzen Sie ein Objekt Ihrer Implementierung in der Zeile 10 der Klasse `main.Dialog`. Die Aufgaben der Methoden sind im Interface dokumentiert. Details der Umsetzung sind Ihnen überlassen. Java bietet u. a. die Klassen `XMLEncoder` und `XMLDecoder` sowie `ObjectInputStream` und `ObjectOutputStream` zur Persistierung an. Der folgende Nutzungsdialog soll möglich sein.

Was wollen Sie?

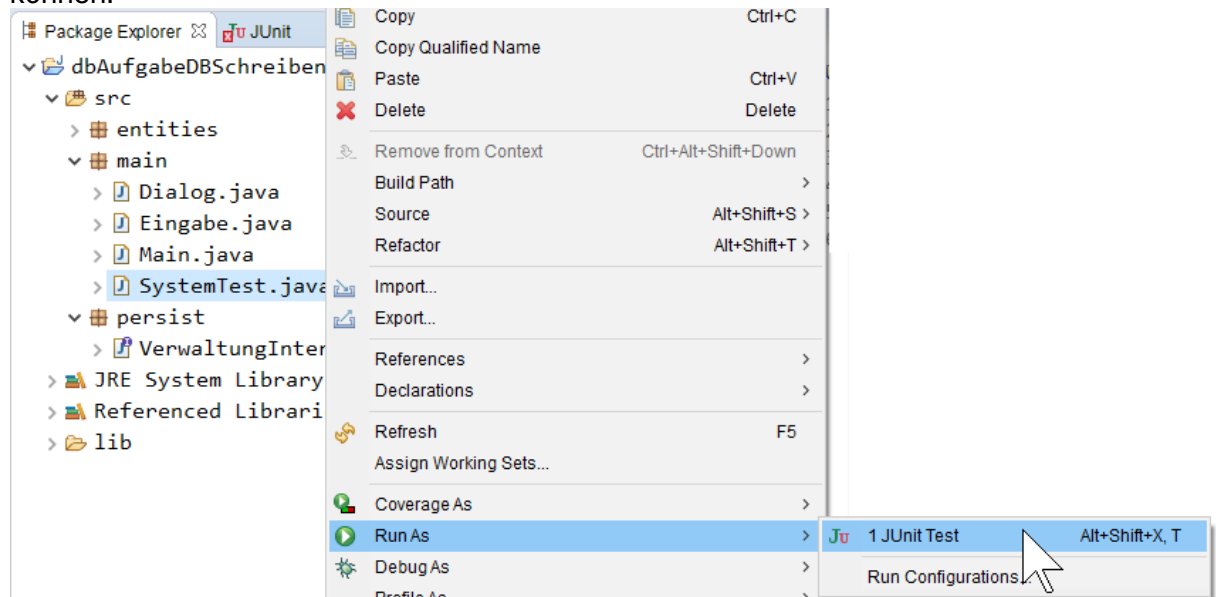
- (0) Programm beenden
- (1) neuen Studi hinzufügen
- (2) alle Studis zeigen
- (3) Namen eines Studi ändern
- (4) Studis eines Fachs zeigen
- (5) Studi löschen
- (6) Daten speichern
- (7) Daten laden

Nutzen Sie wie üblich zur Ausgabe von Studi-Objekten die bereits vorhandene Methode `toString()`. Die Ausgaben sollen wie folgt aussehen.

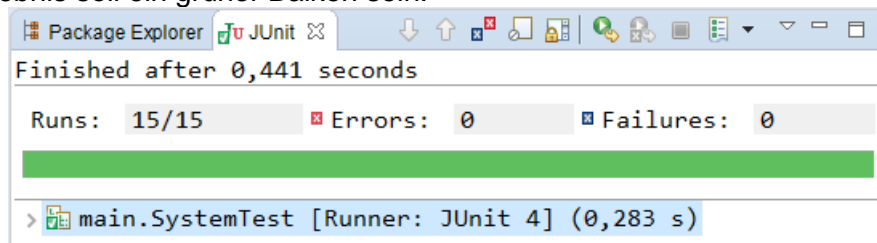
```
2
Studi{mat=48, name=Jana, fach=WI}
Studi{mat=49, name=Esra, fach=TI}
Studi{mat=42, name=Ute, fach=TI}
Studi{mat=43, name=Ulf, fach=TI}
Studi{mat=44, name=Sergej, fach=TI}
Studi{mat=45, name=Ahmad, fach=MI}
Studi{mat=46, name=Jan, fach=MI}
Studi{mat=47, name=Jim, fach=WI}
```

Wird eine vorhandene Matrikelnummer für neue Studierende eingegeben, wird kein neues Objekt erzeugt. Änderungen für nicht existierende Studierende werden ebenfalls verworfen.

Testen Sie Ihre Lösung mit den vorhandenen Tests in `main.SystemTest.java`, die z. B. über einen Rechtsklick auf der Datei und „Run As > JUnit Test“ aufgerufen werden können.



Das Ergebnis soll ein grüner Balken sein.



- c) Lassen Sie das Programm mehrfach parallel als Konsolenprogramm laufen, was passiert mit dem Ergebnis, wenn sich die Arbeitsschritte der Nutzungen überlappen, z. B. ein Datensatz in mehreren Konsolen bearbeitet wird?

Zum Aufruf des Programms nutzen Sie `StartKonsole.bat` aus der SEU und gehen Sie in das Verzeichnis bin Ihres Projekts und starten die Ausführung.

```
C:\Administrator: Eingabeaufforderung - java main.Main
F:\workspaces\eclipseWS\dbAufgabeDBSchreiben\bin>java main.Main

Was wollen Sie?
(0) Programm beenden
(1) neuen Studi hinzufuegen
(2) alle Studis zeigen
(3) Namen eines Studi aendern
(4) Studis eines Fachs zeigen
(5) Studi loeschen
(6) Daten speichern
(7) Daten laden
```