

Übung 01: Semesterplaner

Abgabetermin: 10. 3. 2016, 8:15

Name: _____

Matrikelnummer: _____

Informatik: ☐ G1 (Grimmer) ☐ G2 (Prähofer) ☐ G3 (Prähofer)

WIN: ☐ G1 (Khalil) ☐ G2 (Khalil) ☐ G3 (Schwinger)

Aufgabe	Punkte	abzugeben schriftlich	abzugeben elektronisch	korr.	Punkte
Übung 1	24	Java-Programm, Ausgabe eines Testlaufs	Java-Programm	<input type="checkbox"/>	

Übung 01: Semesterplaner (24 Punkte)

In dieser Übung soll ein Programmsystem bestehend aus mehreren Klassen realisiert werden, mit dem man die Lehrveranstaltungen eines Semesters planen und verwalten kann.

Ein Semester besteht aus mehreren Wochen (z.B. SS 2016 17 Wochen) und jede Woche aus den 7 Wochentagen. An jedem dieser Tage hat man eine Reihe von Terminen von Lehrveranstaltungen, die folgende Daten haben:

- Woche
- Wochentag
- Bezeichnung der LVA
- Beginnzeit
- Dauer

Eine Klasse `SemesterScheduler` dient zur Verwaltung der Termine eines Semesters:

- Ein Semester besteht aus einer bestimmten Anzahl von Wochen,
- jede Woche aus 7 Tagen und
- für jeden Tag soll eine Liste von Terminen von Lehrveranstaltungen nach der Zeit aufsteigend sortiert gespeichert werden können.

Implementieren Sie eine lineare Liste für die sortierte Verwaltung der LVAs eines Tages (Klassen aus der Bibliothek, z.B. `ArrayList`, dürfen nicht verwendet werden). Die LVAs sollen nach der Beginnzeit aufsteigend sortiert gespeichert werden. Verwenden Sie am besten ein zweidimensionales Array für die Tage (Dimension = Anzahl Wochen x 7 Tage pro Woche).

Abbildung 1 zeigt schematisch die Speicherstruktur.

Folgende Funktionen sollen beim `SemesterScheduler` unterstützt werden:

- Definition einer LVA mit eindeutigem Namen, an einem gegebenen Wochentag, einer gegebenen Uhrzeit und einer gegebenen Dauer für alle Wochen des Semesters
- Einfügen einer LVA mit eindeutigem Namen, einer bestimmten Uhrzeit und einer bestimmten Dauer an einem einzigen Wochentag in einer einzigen Woche
- Löschen einer LVA mit einem gegebenen Namen an einem bestimmten Wochentag in einer gegebenen Woche
- Ausgeben der LVAs eines gegebenen Tages in einer gegebenen Woche
- Ausgeben der nächsten LVA eines gegebenen Tages in einer gegebenen Woche nach einer bestimmten Uhrzeit
- Ausgeben aller Semestertermine einer LVA mit gegebenen Namen

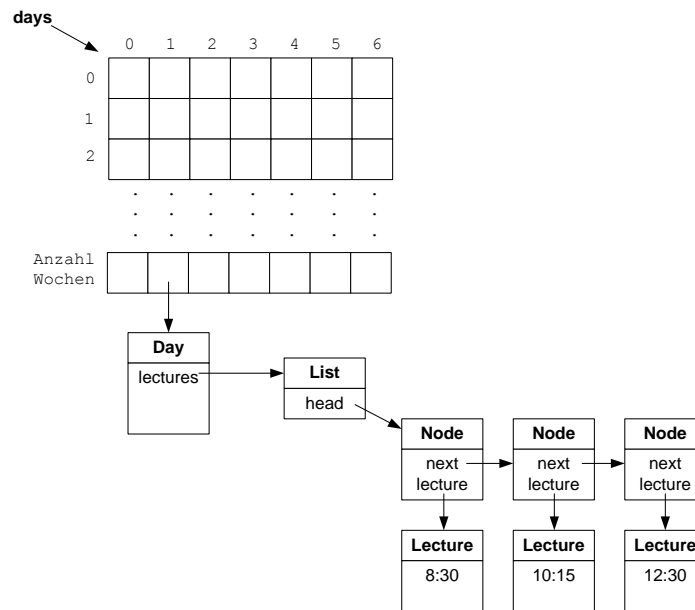


Abbildung 1: Speicherstruktur für Semesterplaner

Beispieldialog:

In einer Programmvorgabe finden Sie einen Benutzerdialog, mit dem Sie Ihr Programm interaktiv testen können. Folgend ist ein Beispiel eines Benutzerdialogs angegeben. Ergänzen Sie die mit

// **TODO:** ...

gekennzeichneten Stellen in der Programmvorgabe mit Aufrufe Ihres SemesterSchedulers.

Semesterplan
=====

Folgende Operationen stehen zur Verfügung:

- D - Definition einer LVA für das Semester
- W - Bestimmen einer Woche
- T - Bestimmen eines Tages der Woche
- A - Ausgeben der LVAs des aktuellen Tages
- N - Ausgeben der nächsten LVA
- E - Einfügen eines LVA beim aktuellen Tag
- L - Löschen einer LVA des aktuellen Tages
- S - Ausgeben der Semestertermine einer LVA
- X - Programm beenden

Bitte Operation auswaehlen: **D**

Eingabe einer LVA

LVA Name: **Rechtsgrundlagen**

Wochentag (1 - 7): **2**

Bitte Zeit eingeben (hh:mm): **10:15**

Dauer: **90**

...

Bitte Operation auswaehlen: **W**

Woche: **3**

Bitte Operation auswaehlen: **T**

Wochentag (1 - 7): **2**

Bitte Operation auswaehlen: **A**

Day: TUESDAY, week 3

Rechtsgrundlagen: week 3, on TUESDAY, from 10:15 to 11:45

Bitte Operation auswaehlen: **T**

Wochentag (1 - 7): **4**

Bitte Operation auswaehlen: **A**

Day: THURSDAY, week 3

SWE2: week 3, on THURSDAY, from 08:30 to 10:00

SWE2UE: week 3, on THURSDAY, from 10:15 to 11:45

Bitte Operation auswaehlen: **N**

Nächste LVA für diesen Tag nach Zeit
Bitte Zeit eingeben (hh:mm): **10:00**
SWE2UE: week 3, on THURSDAY, from 10:15 to 11:45

Bitte Operation auswaehlen: **L**
Löschen einer LVA für diesen Tag
LVA Name: **SWE2UE**

Bitte Operation auswaehlen: **A**
Day: THURSDAY, week 3
SWE2: week 3, on THURSDAY, from 08:30 to 10:00

Bitte Operation auswaehlen: **S**
Termine einer LVA im Semester
LVA Name: **SWE2**
SWE2: week 1, on THURSDAY, from 08:30 to 10:00
SWE2: week 2, on THURSDAY, from 08:30 to 10:00
SWE2: week 3, on THURSDAY, from 08:30 to 10:00
SWE2: week 4, on THURSDAY, from 08:30 to 10:00
SWE2: week 5, on THURSDAY, from 08:30 to 10:00
SWE2: week 6, on THURSDAY, from 08:30 to 10:00
SWE2: week 7, on THURSDAY, from 08:30 to 10:00
SWE2: week 8, on THURSDAY, from 08:30 to 10:00
SWE2: week 9, on THURSDAY, from 08:30 to 10:00
SWE2: week 10, on THURSDAY, from 08:30 to 10:00
SWE2: week 11, on THURSDAY, from 08:30 to 10:00
SWE2: week 12, on THURSDAY, from 08:30 to 10:00
SWE2: week 13, on THURSDAY, from 08:30 to 10:00
SWE2: week 14, on THURSDAY, from 08:30 to 10:00
SWE2: week 15, on THURSDAY, from 08:30 to 10:00
SWE2: week 16, on THURSDAY, from 08:30 to 10:00
SWE2: week 17, on THURSDAY, from 08:30 to 10:00
SWE2: week 18, on THURSDAY, from 08:30 to 10:00

Bitte Operation auswaehlen: **X**

Hinweise und Anleitungen:

- Achten Sie bei dieser Aufgabe auf eine saubere Gestaltung der Schnittstellen der einzelnen Klassen. Überlegen Sie sich genau, was von der Implementierung öffentlich sichtbar und was verborgen sein soll. Überlegen Sie sich für jedes Programmelement, welchen Access-Modifizier Sie verwenden.
- Für Zeiten sollen Sie die bereitgestellte Klasse `Time` im Package `datetime` verwenden. Diese erlaubt mit den Methoden `equals` und `compareTo` den Vergleich zweier `Time`-Objekte. `equals` bestimmt, ob zwei `Time`-Objekte die gleiche Zeit darstellen. `compareTo` liefert einen negativen, positiven oder 0-Wert, je nachdem ob das `Time`-Objekt kleiner, größer oder gleich dem als Parameter übergebenen Objekt ist. Analoges gilt für `Date`-Objekte.
- Bereitgestellt wird auch eine Klasse `DateTimeUtil`, die nützliche Funktionen für den Umgang mit Datum und Zeit enthält. So enthält die Klasse unter anderem Funktionen zur Bestimmung der Anzahl der Tage pro Jahr und Monat, zur Überprüfung der Gültigkeit von Datums- und Zeitangaben und für diverse Umrechnungen.

Vereinfachungen:

- Pro Tag soll es nur eine Lehrveranstaltung mit einem bestimmten Namen geben. Das heißt, Namen von Lehrveranstaltungen sind für einen Tag eindeutig.