

Übung „Design Pattern“

Zu einer Vorlesung gehören

- Ein Dozent (der u.U. wechseln kann)
- Zuhörer (mindestens fünf)
- Inhalte (vereinfacht dargestellt durch eine Folge von Kapitelüberschriften)

Erstellen Sie für jede der folgenden Teilaufgabe Testprogramme.

Aufgabe 1

Erstellen Sie zunächst die Klasse `Lecture` (mit dem Attributen `title` für den Namen der LV).

Geben Sie der Vorlesung ein Attribut `contents`, das eine erweiterbare, geordnete Liste der Kapitelüberschriften enthält. Welchen Datentyp verwenden Sie hierfür am besten?

Mit Hilfe der Methode `addChapter(int, String)` sollen die einzelnen Kapitel zur Vorlesung hinzugefügt werden können. Die Methode `overview` soll die geordnete Liste der Kapitel als String zurückgeben. Leere Kapitel werden als „N.N.“ ausgegeben.

Aufgabe 2

Der Dozent erzeugt eine (tatsächliche) Vorlesung, indem er die Methode `holdLecture` der Klasse `Lecture` aufruft. Der Aufruf soll zu folgender Ausgabe führen:

```
Guten Tag, ich begrüße Sie zur <n>. Vorlesung zum Thema <title>.
```

```
Mein Name ist <name>.
```

```
Unser heutiges Thema ist <contents>.
```

Ein mehrfacher Aufruf soll nacheinander die verschiedenen Inhalte ausgeben. Nach dem letzten Kapitel soll keine Ausgabe mehr erfolgen, bis die Methode `newTerm` (neues Semester) an der Klasse `Lecture` aufgerufen wurde.

Erstellen Sie dazu die Klasse `Professor` und fügen Sie der Klasse `Lecture` die entsprechenden Methoden hinzu.

Aufgabe 3

Die Zuhörer können bei einzelnen Vorlesungen anwesend sein oder auch nicht. Entsprechend bekommen Sie den Inhalt mitgeteilt (oder auch nicht).

Verwenden Sie das Entwurfsmuster Observer, um diese Beziehungen umzusetzen. Gehen Sie wie folgt vor:

1. Leiten Sie die Klasse `Lecture` von `Observable` ab.
Welche Methoden müssen Sie ergänzen?
2. Erstellen Sie eine Klasse `Student`, die das Interface `Observer` implementiert.
Fügen Sie die geforderten Methoden zur Klasse hinzu. Außerdem soll jeder Student einen Namen haben.
3. Bei Benachrichtigung über den Vorlesungsinhalt soll der Student diesen in sein Skript einfügen.
4. Das Skript des Studenten soll per Methode `getScript` als String ausgegeben werden.
5. Geben Sie alternativ das Skript in einer Textdatei mit dem Namen `Lec_Name.txt` aus, wobei `Lec` der Titel der LV und `Name` der Name des Studenten ist.