

Blatt 3

Punktetabelle

| | | |
|----|---|----------|
| 1 | 2 | Σ |
| 10 | 7 | 17 |

1 Aufgabe 1 - Objektorientierte Studenten

Versuchen Sie die bisherigen Konzepte in einer objektorientierten Studentenverwaltung umzusetzen (ähnlich dem Beispiel von Blatt 1). Beginnen Sie mit einer Klasse, die EINEN (!) Studenten abbildet und alle relevanten Daten vereint (Matrikelnummer, Name, Geburtsdatum, Studienkennzahl, Status, ...). Speichern Sie auch die Matrikelnummer explizit als Variable dieser Klasse. Achten Sie insbesondere darauf, den Zugriff der Daten nach außen hin einzuschränken! Studenten sollen per Konstruktor angelegt werden können, wobei Matrikelnummer, Name, Geburtsdatum und Studienkennzahl als Argumente übergeben werden. Implementieren Sie weiters einen Konstruktor, der zusätzlich den Status des Studenten als Argument erhält und erklären Sie anhand dieses Beispiels das Prinzip des Überladens. Können Sie ein Beispiel finden, das ohne ein Überladen von Methoden nicht lösbar ist? Erklären Sie ausführlich!

Erweitern Sie die Klasse Student um Methoden für folgende Funktionalität:

- Die Daten des Studenten ändern sich
- Der Student exmatrikuliert
- Der Student schließt sein Studium ab

Lesen Sie den Eintrag der Klasse Object in der Java API¹. Was macht die Methode *toString*? Implementieren Sie diese Methode für die Klasse Student.

2 Aufgabe 2 - Studentenverwaltung

Erstellen Sie eine Klasse zur Studentenverwaltung, welche mehrere Studenten Objekte in einer Datenstruktur ihrer Wahl speichert. Implementieren Sie Methoden, um einen neuen Studenten anzulegen, einen Studenten zu exmatrikulieren, einen Studenten auszugeben, alle Studenten auszugeben und das Studium eines Studenten zu schließen. Überlegen Sie sich für alle Methoden sinnvolle Signaturen und achten Sie wiederum auf Datenkapselung! Implementieren Sie eine Methode, die als Argument eine beliebig lange Liste an Matrikelnummern bekommt und die entsprechenden Studenten ausgibt (verwenden Sie *varargs*). Testen Sie ihren Code, indem Sie in einer statischen Methode ein Objekt zur Studentenverwaltung anlegen. Dort sollten sie dann auch einige Studenten erstellen, diese verändern, ausgeben, etc.

¹<http://docs.oracle.com/javase/7/docs/api/>