Übungen zur Vorlesung Programmieren 1
FHWS – Fakultät für Informatik und Wirtschaftsinformatik

Übungsblatt 10 (bzv-relevant)

Aufgabe 1: (1 Punkt) (Produktionszähler)

Erweitern Sie die Klasse Auto (Übungsblatt 9 / Aufgabe 1 und 2) um eine static-Variable mit deren Hilfe die Anzahl der produzierten Autos gezählt wird. Schreiben Sie weiterhin eine Klassenmethode **berechneUmsatz(int durchschnittspreis)**, welche anhand des übergebenen durchschnittlichen Verkaufspreises den Bruttoumsatz der Fahrzeugproduktion berechnet und diesen zurück liefert.

Aufgabe 2: (1 Punkt) (Raum)

Schreiben Sie eine Klasse `Raum`. Ein Raum hat eine *Kennung*, die ihn identifiziert (z.B. I.2.1) und kann eine maximale *Anzahl* an Studenten für eine Vorlesung beherbergen. Ferner soll ein Raum (für eine Vorlesung/Übung) *belegbar* sein und darüber Auskunft geben können, ob er gerade belegt ist oder nicht.

Schreiben Sie eine main-Methode, die zwei Räume anlegt, den Raum I.2.1 und den Raum I.2.15, mit der jeweils maximalen Zahl der Studenten, die in den Raum passen. Danach soll die Methode abfragen, ob die Räume belegt sind. Im letzten Schritt soll der Raum I.2.1 belegt werden und nochmals beide Räume befragt werden, ob sie belegt sind. Setzen Sie bei der Aufgabe das Prinzip des **Information Hidings** um!

Hinweis: Ob ein Raum belegbar ist oder nicht, lässt sich über eine bool'sche Variable darstellen.

Aufgabe 3: (1 Punkt) (Raum und Vorlesung)

Erweitern Sie Ihr Programm aus Aufgabe 2, um eine Klasse `Vorlesung`. Eine Vorlesung soll einen *Namen* haben, von einem *Dozenten* gehalten werden und einem *Studiengang* zugeordnet sein.

Wenn ein Raum belegt wird, dann soll er mit einer bestimmten Vorlesung (also einem Objekt der Klasse Vorlesung) belegt werden.

Wenn der Raum nach seiner Belegung gefragt wird, soll der die Vorlesung zurückgeben, mit der er gerade belegt ist. Wenn der Raum frei ist, soll er `null` zurückgeben.

Aufgabe 4: (1 Punkt) (Raum und Studenten)

An einer Hochschule, die in dieser Aufgabe nicht genannt werden möchte, sind alle Studenten mit einer Chipkarte ausgerüstet, die es zu jeder Zeit erlaubt, den Studierenden zu orten.
Diese Information soll benutzt werden, um festzustellen, welche Studenten zu einem bestimmten Zeitpunkt in einem Raum sind.

Ergänzen Sie die Klasse `Raum` (aus Aufgabe 2) so, dass er Studenten erfassen kann und zwar maximal gemäß seiner Kapazität. Immer wenn ein Student einen Raum betritt, soll der Raum den Studenten abspeichern. Wenn der Student den Raum verlässt, soll er den Studenten wieder entfernen.

Vorgehensweise: Ergänzen Sie im Raum als Attribut ein Array von Studenten. Erweitern Sie den Raum, um eine Methode `betreteRaum()` und übergeben Sie beim Aufruf der Methode den Studenten, der den Raum betritt und speichern Sie ihn im Array. Gehen Sie analog für eine Methode `verlasseRaum()` vor.

Schreiben Sie ein Testprogramm, das den Raum mit Studenten belegt, alle Studenten, die zum Zeitpunkt der Abfrage im Raum sind, ausgibt und dann die Studenten wieder den Raum verlassen lässt.