

Übung 7, Lösungsvorschlag



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Aufgabe 1 (Studierendenverwaltung)

Sie möchten ein Programm zur Verwaltung von Studierenden und absolvierten Prüfungen schreiben. Zuerst werden Sie eine Klasse erzeugen, die Prüfungsergebnisse abbildet. Danach werden Sie eine weitere Klasse Leistungsspiegel implementieren, welche basierend auf einem Array verschiedene Prüfungen speichern kann. Als drittes werden Sie eine Klasse implementieren, die Studierende und ihren jeweiligen Leistungsspiegel abbildet. Schließlich werden Sie eine Klasse Studierendenverwaltung schreiben, die das Hauptprogramm (main-Methode) enthält. Gehen Sie dazu wie folgt vor:

- a) Implementieren Sie zunächst eine Klasse mit dem Bezeichner `Klausur`. Diese verfügt über zwei Attribute: ein `String`-Attribut mit dem Bezeichner `name` und ein weiteres `double`-Attribut mit dem Bezeichner `note`. Außerdem soll diese Klasse nun einen Konstruktor erhalten, welcher zwei Parameter erhält. Der erste Parameter soll einen Wert für das Attribut `name` erhalten. Der zweite Parameter enthält einen Wert für eine `note`. Weisen Sie in diesem Konstruktor die übergebenen Werte den entsprechenden Attributen der Klasse `Klausur` zu.
Um diese Klasse zu testen, schreiben Sie nun die Klasse `Studierendenverwaltung` mit einem Hauptprogramm und erzeugen Sie innerhalb der `main`-Methode zwei `Klausur`-Objekte mit den Namen „GdP“ und „GdW“. Lassen Sie anschließend sowohl den Namen als auch die `note` auf der Konsole ausgeben.
- b) Implementieren Sie nun die zweite Klasse mit dem Bezeichner `Leistungsspiegel`. Diese soll im Folgenden in einem Attribut mit dem Bezeichner `Klausuren` mehrere Objekte vom Typ `Klausur` in einem Array speichern. Erstellen Sie noch ein weiteres Attribut `anzahlKlausuren` vom Datentyp `Integer`, das die Zahl der zur Zeit in das Array hinzugefügten `Klausur`-Objekte enthalten soll (also zu Beginn 0).
- c) Erstellen Sie nun für die Klasse `Leistungsspiegel` einen Konstruktor, der einen einzelnen Parameter vom Datentyp `integer` erhält. Dieser soll die maximale Anzahl an möglichen Klausuren im Leistungsspiegel festlegen (also die Länge des Arrays `Klausuren[]`). Erzeugen Sie somit im Konstruktor auf dynamische Art ein neues Array vom Datentyp `Klausur` und verwenden Sie den übergebenen Parameter als Wert zur Definition der Länge des Arrays.
- d) Nun soll die Klasse `Leistungsspiegel` eine Methode `klausurEintragen` erhalten, welche als Parameter ein `Klausur`-Objekt mit dem Bezeichner `k` erhält. Zuerst müssen Sie sicherstellen, dass das Attribut `Klausuren[]` noch einen freien Eintrag hat. Vergleichen Sie also den Wert des Attributs `anzahlKlausuren` mit der Länge des Attributs `Klausuren[]`. Wenn kein freies Element mehr existiert, geben Sie eine Fehlermeldung auf der Konsole aus. Andernfalls speichern Sie das übergebene `Klausur`-Objekt `k` an den nächsten noch freien Index des Attributs `Klausuren[]` (nutzen Sie hierzu das Attribut `anzahlKlausuren`). Danach muss `anzahlKlausuren` um Eins erhöht werden, damit beim wiederholten Aufruf der Methode die nächste Klausur im nächsten Element des Arrays gespeichert wird.
Geben Sie nun durch folgenden Aufruf die so hinzugefügte Klausur auf der Konsole aus:
`System.out.println("Es wurde die " + anzahlKlausuren + ". Klausur " + k.name + " gespeichert.");`

Um diese Klasse nun zu testen, erzeugen Sie innerhalb des Hauptprogramms aus a) ein Leistungsspiegel-Objekt. Wählen Sie hierzu eine Anzahl an Klausuren von 2. Tragen Sie nun die in a) bereits angelegten Klausur-Objekte in den Leistungsspiegel ein. Erzeugen Sie nun ein drittes Klausur-Objekt und tragen dieses ebenfalls in den Leistungsspiegel ein. Die Ausgabe sollte wie folgt aussehen:

Es wurde die 1. Klausur GdP gespeichert.

Es wurde die 2. Klausur GdW gespeichert.

Fehler: Es kann keine weitere Klausur eingetragen werden.

- e) Schreiben Sie nun die Klasse `Studierender`, die über die Attribute `matrikelnr` (int), `vorname` (String), `nachname` (String) und einen `leistungsspiegel` (vom eben implementierten Datentyp `Leistungsspiegel`) verfügt. Außerdem soll die Klasse ein Klassenattribut mit dem Bezeichner `anzahlStudierende` erhalten (wählen Sie wieder den Modifier `public`).
- f) Fügen Sie auch der Klasse `Studierender` einen Konstruktor hinzu. Der Name eines Studierenden soll immer beim Erstellen gesetzt werden. Daher erwartet der Konstruktor zwei Parameter vom Typ `String`. Diese werden den Attributen `vorname` und `nachname` zugewiesen. Die Matrikelnummer soll automatisch gesetzt werden, sodass jedes neue `Studierender`-Objekt eine fortlaufende Matrikelnummer zugewiesen bekommt. Die erste Matrikelnummer ist 1000. Nutzen Sie zur Berechnung der zu vergebenen Matrikelnummer das statische Klassenattribut `anzahlStudierende`. Vergessen Sie nicht, dieses Attribut anschließend um Eins zu erhöhen, damit die folgenden Kommilitonen die nächst größere Matrikelnummer zugewiesen bekommen.
- g) Um nun auch die Klasse `Studierender` zu testen, erzeugen Sie im Hauptprogramm zwei `Studierender`-Objekte `s1` und `s2` und weisen ihnen entsprechend Vorname und Nachname zu. Anschließend erzeugen Sie ebenfalls ein zweites `Leistungsspiegel`-Objekt und tragen das Ergebnis für eine Klausur ein. Weisen Sie nun beiden `Studierender`-Objekten jeweils ein eigenes `Leistungsspiegel`-Objekt zu. Geben Sie schließlich auf der Konsole mit folgenden Befehlen die Informationen zu Ihren zwei Studierenden aus. Prüfen Sie ihr Ergebnis anhand Ihrer selbst gewählten Angaben (Vornamen, Nachnamen, Anzahl an eingetragenen Klausuren).

```
System.out.println(s1.matrikelnr + " hat " +  
s1.leistungsspiegel.anzahlKlausuren + " Klausuren absolviert");
```

```
System.out.println(s2.matrikelnr + " hat " +  
s2.leistungsspiegel.anzahlKlausuren + " Klausuren absolviert");
```

Die Ausgabe sollte wie folgt lauten:

```
1000 hat 2 Klausuren absolviert
```

```
1001 hat 1 Klausuren absolviert
```

Lösung Aufgabe 1

```
1  // Aufgabe a
2  public class Klausur {
3      public String name;
4      public double note;
5
6      public Klausur(String na, double no) {
7          name = na;
8          note = no;
9      }
10 }
```

```
1  // Aufgabe b - d
2  public class Leistungsspiegel {
3      public Klausur[] klausuren;
4      public int anzahlKlausuren = 0;
5
6      public Leistungsspiegel(int anzahl) {
7          klausuren = new Klausur[anzahl];
8      }
9
10     public void klausurEintragen(Klausur k) {
11         if (anzahlKlausuren >= klausuren.length) {
12             System.out.println("Fehler: Es kann keine weitere
13             Klausur eingetragen werden.");
14         } else {
15             klausuren[anzahlKlausuren] = k;
16             anzahlKlausuren++;
17             System.out.println("Es wurde die " + anzahlKlausuren +
18             ". Klausur " + k.name + " gespeichert.");
19         }
20     }
21 }
```

```
1  // Aufgabe e + f
2  public class Studierender {
3      public int matrikelnr;
4      public String vorname;
5      public String nachname;
6      public Leistungsspiegel leistungsspiegel;
7
8      public static int anzahlStudierende = 0;
9
10     public Studierender(String vn, String nn) {
11         vorname = vn;
12         nachname = nn;
13         matrikelnr = 1000 + Studierender.anzahlStudierende;
14
15         Studierender.anzahlStudierende++;
16     }
17 }
```

```
1  public class Studierendenverwaltung {
2      public static void main(String[] args) {
3          // Aufgabe a)
4          Klausur GdP = new Klausur("GdP", 1.7);
5          Klausur GdW = new Klausur("GdW", 2.3);
6
7          //Aufgabe d)
8          Leistungsspiegel spieglein1 = new Leistungsspiegel(2);
9
10         spieglein1.klausurEintragen(GdP);
11         spieglein1.klausurEintragen(GdW);
12
13         Klausur mathe = new Klausur("Mathe", 2.0);
14         spieglein1.klausurEintragen(mathe);
15
16         // Aufgabe g)
17         Studierender s1 = new Studierender("Chris", "2000");
18         Studierender s2 = new Studierender("Jin", "Gerlach");
19
20         Leistungsspiegel spieglein2 = new Leistungsspiegel(1);
21         Klausur mathe2 = new Klausur("Mathe2", 1.7);
22         spieglein2.klausurEintragen(mathe2);
23
24         s1.leistungsspiegel = spieglein1;
25         s2.leistungsspiegel = spieglein2;
26
27         System.out.println(s1.matrikelnr + " hat " +
28             s1.leistungsspiegel.anzahlKlausuren + " Klausuren absolviert");
29         System.out.println(s2.matrikelnr + " hat " +
30             s2.leistungsspiegel.anzahlKlausuren + " Klausuren absolviert");
31     }
32 }
```