Prof. Dr. Franz Schweiggert
Christoph Ott
Institut für Angewandte Informationsverarbeitung
12. Juni 2008
Blatt 7

Einführung in die Programmierung (SS 2008)

Abgabetermin: 19. Juni 2008

1 Arrays & Methoden

Mit Arrays kann man eine ganze Menge machen. Damit man nicht für jeden Array, den man einlesen, ausgeben, sortieren, etc. möchte den Code wieder komplett neu schreiben (bzw. kopieren) muss, sind hier Methoden besonders geschickt.

Auf diesem Übungsblatt sollt Ihr Euren Notenspiegel in einem Array speichern und dann verschiedene Methoden für diesen Array aufrufen.

1.1 Kleine Schleife zum Einlesen der Matrikelnummer (2P)

Schreibt eine kleine Schleife, die so lange die Matrikelnummer eines Studenten einliest und in einer geeigneten Variablen speichert bis der Benutzer irgendetwas eingibt, das nicht als Matrikelnummer interpretiert werden kann (also keine ganze Zahl ist). Auf dem Bildschirm kann das etwa folgendermaßen aussehen:

```
Geben Sie bitte Ihre Matrikelnummer ein!
123456

Geben Sie bitte Ihre Matrikelnummer ein!
654321

Geben Sie bitte Ihre Matrikelnummer ein!
schokofondue
```

Hinweis: Sollte jemand an dieser Schleife scheitern, so möge er/sie doch die folgenden Teilaufgaben einfach ohne der Schleife bearbeiten.

1.2 Einlesen der Noten und Ausgeben des Notenspiegels (2P)

Für jede gültige Matrikelnummer soll nun nach der Anzahl bereits absolvierter Prüfungen gefragt werden und ein Array von dieser Größe angelegt werden. Dieser (noch leere) Array soll nun einer Methode *readArray* übergeben, die den Benutzer nach seinen bisherigen Noten fragt und den Array füllt. Danach soll im Hauptprogramm eine Methode *printArray* aufgerufen werden, der der gefüllte Array übergeben wird und die diesen elementweise ausgibt. Auf dem Bildschirm kann das etwa folgendermaßen aussehen:

```
Geben Sie bitte Ihre Matrikelnummer ein!
123456
Wieviele Pruefungen hab Sie bereits absolviert?
4

Geben Sie bitte Ihre Noten ein und bestaetigen Sie jeweils mit Return!
Note 1: 2.0
Note 2: 3.3
Note 3: 1.7
Note 4: 2.3

Notenspiegel:
Note 1: 2.0
Note 2: 3.3
Note 3: 1.7
Note 4: 2.3

Geben Sie bitte Ihre Matrikelnummer ein!
karnickelzuechterverein
```

Tipp: Bei der Bearbeitung dieser Teilaufgabe kann die Beispielaufgabe aus der Vorlesung *RA.java* hilfreich sein.

1.3 Suchen nach bester und schlechtester Note (2P)

Nun soll Euer Programm um zwei weitere Methoden erweitert werden. Die erste dieser Methoden soll den Array durchlaufen und das kleinste Element (also die beste Note) an das Hauptprogramm zurückliefern, das diese dann ausgeben soll. Die zweite Methode soll das selbe mit dem größten Element, also der schlechtesten Note, tun. Auf dem Bildschirm kann das etwa folgendermaßen aussehen:

```
Geben Sie bitte Ihre Matrikelnummer ein!
123456
Wieviele Pruefungen hab Sie bereits absolviert?
3

Geben Sie bitte Ihre Noten ein und bestaetigen Sie jeweils mit Return!
Note 1: 2.0
Note 2: 1.3
Note 3: 2.7

Notenspiegel:
```

```
Note 1: 2.0
Note 2: 1.3
Note 3: 2.7

Beste Note: 1.3

Schlechteste Note: 2.7

Geben Sie bitte Ihre Matrikelnummer ein!
3.1415
```

1.4 Bilden des Notendurchschnitts (2P)

Jetzt kommen wieder zwei Methoden hinzu. Die erste der beiden soll den Notdendurchschnitt der absolvierten Prüfungen berechnen. Dazu soll die Methode den Array entgegennehmen, ihn durchlaufen und den Notendurchschnitt zurückgeben. Die zweite Methode soll einen gewichteten Notendurchschnitt bilden, der jede Note mit der Anzahl an Leistungspunkten, die es für dieses Modul gibt, multipliziert und anschließend durch die Summe an Leistungspunkten dividiert. Dazu müsst Ihr natürlich in der Methode *weightedAverage* zuerst die Leistungspunte jeder Prüfung abfragen. Auf dem Bildschirm kann das etwa folgendermaßen aussehen:

```
Geben Sie bitte Ihre Matrikelnummer ein!
123456
Wieviele Pruefungen hab Sie bereits absolviert?
3

Geben Sie bitte Ihre Noten ein und bestaetigen Sie jeweils mit Return!
Note 1: 1
Note 2: 2
Note 3: 2

Notenspiegel:
Note 1: 1.0
Note 2: 2.0
Note 3: 2.0

Beste Note: 1.0

Schlechteste Note: 2.0

Durchschnitt: 1.6

Geben Sie bitte die Leistungspunkte der einzelnen Noten ein und bestaetigen Sie jeweils
Note 1: 1.0, LP: 4
Note 2: 2.0, LP: 2
Note 3: 2.0, LP: 2
Gewichteter Durchschnitt: 1.5

Geben Sie bitte Ihre Matrikelnummer ein!
feuerstein
```

Sowohl Durchschnitt als auch gewichteter Durchschnitt können natürlich beliebig viele Nachkommastellen besitzen. Bitte beachtet beim Formatieren, dass Notendurchschnitte niemals gerundet, sondern immer abgeschnitten werden.

1.5 Noten als Text (2P)

Im Zeugnis erscheinen die Noten oft auf ganze Noten gerundet und als Text. Schreibt eine Methode die den Noten-Array entgegennimmt, dann jede Note auf eine ganze Zahl rundet und ihr den entsprechenden Noten-Text ("sehr gut", "gut", "befriedigend", "ausreichend" oder "ungenügend") zuweist. Der mit den Noten-Texten gefüllte Array soll zum Schluss augegeben werden. Auf dem Bildschirm kann das etwa folgendermaßen aussehen:

```
Geben Sie bitte Ihre Matrikelnummer ein!
123456
Wieviele Pruefungen hab Sie bereits absolviert?
5

Geben Sie bitte Ihre Noten ein und bestaetigen Sie jeweils mit Return!
Note 1: 3.7
Note 2: 1.3
Note 3: 2
Note 4: 5
Note 5: 2.7

Notenspiegel:
Note 1: 3.7
Note 2: 1.3
Note 3: 2.0
Note 4: 5.0
Note 5: 2.7

Beste Note: 1.3

Schlechteste Note: 5.0

Durchschnitt: 2.9

Geben Sie bitte die Leistungspunkte der einzelnen Noten ein und bestaetigen Sie jeweils
Note 1: 3.7, LP: 4
Note 2: 1.3, LP: 2
Note 3: 2.0, LP: 6
Note 4: 5.0, LP: 6
```

```
Note 5: 2.7, LP: 2
Gewichteter Durchschnitt: 3.1

Notenspiegel:
Note 1: ausreichend
Note 2: sehr gut
Note 3: gut
Note 4: ungenuegend
Note 5: befriedigend

Geben Sie bitte Ihre Matrikelnummer ein!
true
```

Ein paar Tipps hierzu:

- Mit *Math.round()* könnt Ihr eine Zahl runden. Allerdings muss die Zahl die in den runden Klammern angegeben werden muss vom Typ *float* sein. Deshalb müsst Ihr an dieser Stelle Eure *double* explizit nach *float* konvertieren.
- Da Ihr in einem double-Array keine Strings speichern könnt, werdet Ihr wohl nicht darum herum kommen in Eurer Methode einen zweiten Array (von Strings) anzulegen und diesen dann an das Hauptprogramm zurückzugeben.
- Zur Zuweisung der gerundeten Zahl an den Noten-Text empfiehlt sich wieder eine *switch*-Anweisung.

Viel Erfolg!