

Präsenzaufgaben 1

25.03.2019

Die Lösung der Aufgaben wird am Ende der Übung von Ihnen vorgestellt.

Aufgabe 1:

Ihr Chef gibt Ihnen die Aufgabe, ein Feld von Zahlen (`Zahl[]`) zu erzeugen, deren Elemente jeweils

- einen Bruch (aus zwei Integer-Zahlen)
- oder die Wurzel einer Integer-Zahl

aufnehmen können. Ihr Vorgänger hat dieses Problem bereits gelöst. Die entsprechende Klasse `Zahl` können Sie sich von der Kursseite herunterladen. Der Code läuft auch, aber Ihr Chef hat das Gefühl, dass auf Fehlerbehandlung und Erweiterbarkeit eher wenig Wert gelegt wurde.

- (a) Was ist das Problem bezüglich Erweiterbarkeit?
- (b) Nennen Sie 3 Beispiele für fehlerhafte Aufrufe.
- (c) Schreiben Sie das Programm (einschließlich Testprogramm) neu und benutzen Sie einen guten objektorientierten Ansatz. Achten Sie auch auf Fehlerbehandlung.

Aufgabe 2:

- (a) Schreiben Sie eine Klasse `Tripel`, die 3 `int`-Werte zusammenfasst. Die Klasse benötigt lediglich einen Konstruktor zum Setzen der drei Werte und eine `toString`-Methode zur formatierten Ausgabe. Die Benutzung soll wie folgt aussehen:

```
Tripel t = new Tripel(1,2,3);  
System.out.println(t); // Ergibt (1/2/3)
```

- (b) Benutzen Sie die Klasse `Tripel` für die folgende Aufgabe: Schreiben Sie eine Funktion

```
public static ArrayList<Tripel> getTripel(int n)
```

die alle Zahlentripel a, b, c zurückgibt, die gleichzeitig die folgenden 4 Eigenschaften erfüllen:

- $a, b, c \in \mathbb{N}$
- $a < b < c$
- $a^2 + b^2 = c^2$
- $a + b + c = n$ (n ist der Übergabeparameter).

Testfälle:

$n=30 \rightarrow (5/12/13)$

$n=252 \rightarrow (36/105/111), (56/90/106), (63/84/105)$

Aufgabe 3:

Schreiben Sie eine Funktion, die die Werte eines `int`-Felds auf dem Bildschirm ausgibt, wobei der Benutzer die Möglichkeit hat, die Werte anhand eines Kriteriums zu filtern. Bearbeiten Sie dazu die folgenden Punkte:

- (a) Schreiben Sie die Funktion selbst. Die Kopfzeile soll das folgende Aussehen haben:

```
public static void print(int[] liste, Filter filter)
```

- (b) Die Klasse `Filter` soll ein Interface sein. Entwerfen und schreiben Sie das Interface.

- (c) Schreiben Sie die drei folgenden Klassen, die das Interface implementieren:

```
//laesst nur Primzahlen durch
public class PrimeFilter implements Filter

//laesst nur Zahlen mit 2 Ziffern durch,
//also 10 bis 99 und -10 bis -99
public class TwoDigitsFilter implements Filter

//laesst keine Zahlen durch, die in der Folge
//bereits vorkamen.
public class UniqueFilter implements Filter
```

Ihr Code soll am Ende folgendermaßen benutzbar sein:

```
int[] test = {5,9,2,7,2,6,7,12,5,43,4};
print(test, new PrimeFilter());
//ergibt die Zahlen 5,2,7,2,7,5,43 (nur Primzahlen)
print(test, new TwoDigitsFilter());
//ergibt die Zahlen 12,43 (nur Zahlen mit zwei Ziffern)
print(test, new UniqueFilter());
//ergibt die Zahlen 5,9,2,7,6,12,43,4 (jede Zahl nur beim ersten
//Vorkommen)
```

Die Formatierung der Ausgabe ist dabei unwichtig.