Prof. Dr. Christoph Bockisch
MSc Steffen Dick
Fachbereich Mathematik und Informatik
AG Programmiersprachen und -werkzeuge

Philipps Universität Marburg

Übungen zur Vorlesung
Objektorientierte Programmierung: Wintersemester 2021/2022

Nr. 5, Abgabe bis 29.11.2021

Aufgabe 5.1: Max und Minritz — 5 Punkte

In dieser Aufgabe sollen Sie verschiedene Operationen für Arrays implementieren. Sie dürfen jeden Schleifentyp (for, while, foreach) sowie Rekursion nur in einer der Teilaufgaben benutzen.

Implementieren Sie die folgenden Methoden:

a) `void testAll()`, die alle unten aufgeführten Methoden testet. Schreiben Sie Ihre Tests, bevor Sie ihre die anderen Methoden implementieren.

b) `float getMinimum(float[] numbers)`, die das Minimum der übergebenen Zahlen zurückgibt.

c) `float getMaximum(float[] numbers)`, die das Maximum der übergebenen Zahlen zurückgibt.

d) `float calculateAverage(float[] numbers)`, die den berechneten Durchschnittswert der übergebenen Zahlen zurückgibt.

e) `boolean isSorted(float[] numbers, boolean ascending)`, die überprüft, ob die Zahlen entweder in aufsteigender (ascending == true) oder absteigender (ascending == false) Reihenfolge sortiert sind.

Aufgabe 5.2: Lauf, Forest, lauf! 7 Punkte

Das Fitnessband eines Läufers speichert jede Sekunde die GPS-Koordinate, an der sie sich gerade befindet. Eine GPS-Koordinate besteht aus drei Punkten, einem X-Wert, einem Y-Wert und einem Z-Wert. Sie können davon ausgehen, dass es sich bei diesen Werten um Meter handelt.

Das Fitnessband speichert die Werte auf einer kleinen Chip-Karte, von der sie als Array übernommen werden können. Dieses Array hat die folgende Form:

```
double[] gps = {x1,y1,z1,x2,y2,z2,...};
```

Schreiben Sie Tests für die folgenden Methoden, die die GPS-Koordinaten analysieren, und implementieren Sie diese Methoden anschließend:

a) `double distance(double[] gps)`, die die zurückgelegte Strecke approximiert.
b) `double velocity(double[] gps)`, die die Durschschnittsgeschwindigkeit des Läufers zurückgibt (m/s).
c) `double maxVelocity(double[] gps)`, die die maximale Geschwindigkeit berechnet, die der Läufer zurückgelegt hat (gemessen zwischen zwei Punkten).
d) `double[] partialGPS(double[] gps, double[] start, double[] end)`, die einen Teil der GPS-Koordinaten, gestartet bei `start` und beendet mit `end`, zurückgibt (`start` und `end` haben eine Länge von 3).

   Ist einer der beiden Werte **`null`**, kann nicht gefunden werden oder liegt `end` vor `start`, soll das originale Array unverändert zurückgegeben werden.
e) Testen Sie Ihre Methoden ausführlich.

Hinweis: Um eine Distanz im dreidimensionalen Raum zu berechnen, können Sie die folgende Gleichung verwenden:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Hinweis: Sie dürfen `Math.pow`, `Math.sqrt` und `Arrays.copyOfRange(oldArray, startIndex, endIndex)` (endIndex ist exklusiv) beim Lösen dieser Aufgaben verwenden.

Für Ihre Tests dürfen Sie das untenstehende Array verwenden.

```
1 {-20.0, 0.0,200.0, -18.5, -0.647,200.577, -16.85, -1.237,
  ↪ 201.16, -15.035, -1.763, 201.739, -13.038, -2.219,
  ↪ 202.299, -10.842, -2.599, 202.824, -8.426, -2.894,
  ↪ 203.289, -5.769, -3.096, 203.667, -2.846, -3.195, 203.918,
  ↪ 0.369, -3.182, 203.998, 3.861, -3.205, 203.85, 7.284,
  ↪ -3.176, 203.469, 10.638, -3.209, 202.868, 13.926, -3.175,
  ↪ 202.06, 17.147, -3.209,201.059, 20, -3.174, 199.877}
```