

Methodisches Programmierung

Übungen zur Vorlesung

Übungsblatt 4 (zu besprechen in Vorlesungswoche 5)

Beachten Sie diese Anweisungen bei allen folgenden Programmierübungen:

- *Achten Sie bei allen Funktionen, Werten und Variablen auf eine möglichst gute Benennung.*
- *Schreiben Sie für alle Methoden Tests. Schreiben Sie diese bevor Sie die Methoden implementieren.*
- *Dokumentation: Ist der Name einer Methode nicht selbsterklärend, sollten Sie zunächst überlegen, ob Sie nicht einen besseren Namen vergeben können (siehe Punkt 1). Wenn das nicht möglich ist, erklären Sie möglichst präzise in einem Code-Kommentar, was die Methode tut.*
- *Fragen Sie in der A-Übung nach, wenn die Aufgabenstellung missverständlich ist!*

1 Arrays

1. Schreiben Sie eine Funktion

```
createRandomIntArray(size : Int) : IntArray
```

die ein `IntArray` der Länge `size` mit zufälligen `Int`-Werten erzeugt. Einen zufälligen `Int`-Wert können Sie mit `Random.nextInt()` erzeugen. (Bei Tests überprüfen Sie nur die Länge, nicht den Inhalt des Ergebnis-Arrays.)

2. Schreiben Sie eine Funktion,

```
sumThrowsWithTwoDice(numberOfSides : Int) : IntArray
```

die ein Array mit allen möglichen Summen der Werte eines Wurfs mit zwei Würfeln füllt, also `[2, 3, 3, ..., 11, 11, 12]` (ergibt sich aus den Würfeln `[(1, 1), (1, 2), (2, 1), ..., (5, 6), (6, 5), (6, 6)]` mit einem Würfel mit sechs Seiten). Die Funktion soll es ermöglichen, die Anzahl der Seiten der beiden Würfel als Parameter zu übergeben.

3. Implementieren Sie eine Funktion

```
swap(a : IntArray, index1 : Int, index2 : Int) : Unit
```

Welche bei einem `IntArray` die Einträge an `index1` und `index2` vertauscht.

4. Implementieren Sie eine Funktion

```
shuffle(a : IntArray, numberOfSwaps : Int) : Unit
```

welche den Inhalt eines `IntArray`s vermischt. Zum Vermischen sollen mehrfach (`numberOfSwaps` oft) die Werte an zwei zufällig ermittelten Indizes getauscht werden.

5. Schreiben Sie eine Funktion, die den Durchschnitt der Werte in einem `IntArray` ermittelt und diese als `double`-Wert zurückliefert.**6. Implementieren Sie eine Funktion, die den Median der Werte eines `IntArray`s berechnet.****7. Implementieren Sie eine Funktion**

```
std(a : IntArray) : Double
```

die die Standardabweichung der Werte eines `IntArray`s berechnet. (Researchieren Sie online, falls Sie nicht wissen sollten, was die Standardabweichung ist). Was ist `std(sumThrowsWithTwoDice(6))`?

8. Implementieren Sie eine Funktion, die überprüft, ob ein `IntArray` sortiert ist.

```
isSorted(a : IntArray) : Boolean
```

9. Implementieren Sie eine Funktion

```
sortTwoRandomIndices(a : IntArray) : Unit
```

die zufällig zwei Indizes in einem gegebenen `IntArray` auswählt und die Einträge vertauscht, wenn der Wert mit dem kleineren Index größer ist als der Wert mit dem größeren Index.

10. Implementieren Sie eine Funktion

```
randomSwapSort(a : IntArray) : Unit
```

die auf einem gegebenen `IntArray` die Funktion `sortTwoRandomIndices` solange aufruft, bis das `IntArray` sortiert ist.

11. Wir wollen nun messen, wie lange der Sortieralgorithmus `randomSwapSort` für Arrays verschiedener Längen benötigt.

Implementieren Sie dazu eine Funktion

```
benchmarkRandomSwapSort(arraySizes : IntArray) :  
    LongArray
```

die für jeden Eintrag x in `arraySizes` ein zufälliges `IntArray` mit der Größe x erzeugt und Funktion `randomSwapSort` darauf aufruft. Das Ergebnis-Array soll die Ausführungszeiten in Millisekunden zurückgeben. Verwenden Sie dazu die Kotlin-Funktion `measureTime`¹.

Messen Sie die Ausführungszeiten für eine Reihe aufeinanderfolgender Array-Größen, z.B. 5, 6, 7, 8, ... und tragen Sie die Ergebnisse in einen Plot (Graph) ein. Diesen Plot können Sie per Hand oder z.B. mit Excel erzeugen. Können Sie einen Zusammenhang zwischen der Arraygröße und der Laufzeit von `randomSwapSort` erkennen?

Optionale Erweiterung: Erweitern Sie `benchmarkRandomSwapSort` so, dass für jeder Eintrag in `arraySizes` die Messung mehrfach wiederholt wird; speichern Sie dann die Durchschnittsdauer der Ausführungszeiten im Ergebnis-Array.

¹<https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.time/measure-time.html>