

Einführung in die Programmierung
WS 2014/15

Übungsblatt 9: Imperative Programmierung mit Arrays

Besprechung: 17./19.12.2014 und 22.12.2014

Ende der Abgabefrist: Dienstag, 16.12.2014 14:00 Uhr.

Hinweise zur Abgabe:

Geben Sie bitte Ihre gesammelten Lösungen zu diesem Übungsblatt in einer Datei `loesung09.zip` unter <https://uniworx.ifi.lmu.de> ab.

Alle Code-Aufgaben müssen als `.java` abgegeben werden. PDF-code wird nicht akzeptiert. Nicht kompilierbare Abgaben werden mit 0 Punkten bewertet.

Aufgabe 9-1 *Imperative Programmierung und Arrays* **1+1+2 Punkte**

Implementieren Sie die folgenden Berechnungen in der Datei `ArrayProgrammierung.java` und erstellen Sie für jede Unteraufgabe eine Methode. Sie können bei dieser Aufgabe, sowie bei allen zukünftigen Aufgaben, sofern nicht anders spezifiziert, eine iterative oder eine rekursive Lösung implementieren.

- (a) Berechnen Sie für ein `int`-Array die Summe seiner Elemente. Ein Beispiel:

$$\text{summe}([7, 4, 8, 42, 13]) = 74$$

- (b) Berechnen Sie für ein `int`-Array das kleinste Element. Ein Beispiel:

$$\text{min}([7, 4, 8, 42, 13]) = 4$$

- (c) Berechnen Sie für ein `int`-Array dessen Palindrom, das heißt das Array in umgekehrter Reihenfolge.

$$\text{palindrom}([7, 4, 8, 42, 13]) = [13, 42, 8, 4, 7]$$

Aufgabe 9-2 *Klausurpunkte* **4 Punkte**

In der (unvollständigen) Datei `Klausurpunkte.java` werden die Klausurpunkte von Studenten einer virtuellen Vorlesung mit 42 Studenten und acht Klausuraufgaben verwaltet. Für jeden Studenten gibt es ein Array vom Typ `int` (keine halben Punkte) mit den erreichten Punkten zu jeder Aufgabe. Die Punkte aller Studenten für alle Aufgaben werden in einer globalen Variable `punkte` vom Typ `int[][]` gespeichert.

Dieses zwei-dimensionale Array `punkte` enthält also für jeden Studenten s und für jede Aufgabe a die Punktezahl die Student s bei Aufgabe a erreicht hat. Das heißt, der Ausdruck `punkte[s]` gibt ein Array vom Typ `int[]` mit den von Student s erreichten Punkte zu jeder Aufgabe zurück. Ergänzen Sie in der Datei `Klausurpunkte.java` die Methoden `setPunkte`, `getPunkte`, `gesamtPunktezahl` und `durchschnittsPunktezahl`, die in der Datei `Klausurpunkte.java` dokumentiert sind.

Aufgabe 9-3 *Medoid***0 Punkte**

- (a) Schreiben Sie in einer Datei `Medoid.java` eine Methode `medoid`, die für ein Array vom Typ `double[]` diejenige Zahl zurückgibt, die dem Mittelwert aller Zahlen am nächsten liegt.
- (b) Erweitern Sie Ihr Programm um eine entsprechende `main`-Methode, so dass zum Testen obiger Methode `medoid` beim Aufruf der Klasse `Medoid` über die Kommandozeile eine Reihe von Zahlen übergeben werden kann und das Ergebnis der Methode `medoid` auf die Kommandozeile ausgegeben wird.

Beispiel: Der Aufruf

```
java Medoid 2.4 5.1 7 4 14.0 8
```

```
gibt
```

```
7.0
```

auf der Kommandozeile aus.

Hinweis: In der Klasse `java.lang.Double` finden Sie eine geeignete Methode, um einen `String` in die entsprechende Zahl vom Typ `double` zu konvertieren. Sie können auch auf die API-Spezifikation zurückgreifen: <http://docs.oracle.com/javase/7/docs/api/index.html>.

Aufgabe 9-4 *Parameterübergabe***4 Punkte**Gegeben sei die Java-Klasse `Parameteruebergabe`.

```
public class Parameteruebergabe {

    static String a = "A";

    static String b = "B";

    static String[] ab = {a, b};

    public static void swap(String s0, String s1) {
        String temp = s0;
        s0 = s1;
        s1 = temp;
    }

    public static void swap(String[] s) {
        String temp = s[0];
        s[0] = s[1];
        s[1] = temp;
    }

    public static void proverb() {
        System.out.println("Wer " + a + " sagt, muss auch " + b + " sagen!");
        System.out.println("Also muss auch " + ab[0] + " sagen, "
            + "wer " + ab[1] + " sagt?");
    }

    public static void main(String[] args) {
        proverb();
        swap(a, b);
        proverb();
        swap(ab);
        proverb();
    }
}
```

Welche Ausgabe erzeugt das Programm? Begründen Sie Ihre Antwort ausführlich. Geben Sie Ihre Lösung in einer Datei `Parameteruebergabe.txt` ab.

Aufgabe 9-5 *Primfaktorisierung***0 Punkte**

Die Primfaktorzerlegung gibt für eine Zahl $n \in \mathbb{N}$ die Multimenge ihrer Primfaktoren an, d.h. eine Multimenge P von Faktoren $p_i \in P$ so, dass gilt:

1. $n = \prod_{p_i \in P} p_i$
2. Für alle $p_i \in P$ gilt: p_i ist eine Primzahl.

Die Zahl 12 z.B. lässt sich so faktorisieren: $12 = 2 \cdot 2 \cdot 3$. Diese Zerlegung in Primfaktoren ist eindeutig.

Ein einfacher Algorithmus hierzu testet für einen Teiler p , ob die Zahl n durch p teilbar ist. Wenn das der Fall ist, ist p der erste Primfaktor von n . Anschließend werden die Primfaktoren von $\frac{n}{p}$ bestimmt. Andernfalls wird als Teiler $p+1$ getestet, solange $p+1$ nicht größer ist als n . Der erste Teiler, der getestet wird, ist 2, die kleinste Primzahl. Für diese Aufgabe können die Datei `Primfaktoren.java` als Vorlage verwenden.

- (a) Implementieren Sie diesen Algorithmus in einer statischen Methode `printPrimfaktoren` in einer Datei `Primfaktoren.java`, die für eine gegebene Zahl vom Typ `int` einen String auf die Kommandozeile ausgibt, der die Primfaktorisierung der Zahl als arithmetischen Ausdruck darstellt.

Beispiel: Wird die Methode mit dem Parameter 12 aufgerufen, dann soll sie $12 = 2 * 2 * 3$ ausgeben.

- (b) Unter Verwendung der Methode `siebDesEratosthenes`, die Sie in der Datei `Primfaktoren.java` finden, können Sie leicht eine Methode `printGesiebtePrimfaktoren` implementieren, die das gleiche Ergebnis liefert wie `printPrimfaktoren`, aber als Teiler nur Primzahlen testet.

- (c) Überlegen Sie, welche der beiden Methoden effizienter ist (also weniger Speicherplatz und/oder Rechenzeit) benötigt. Theoretische Grundlagen hierzu folgen in der Vorlesung zu einem späteren Zeitpunkt, aber wenn Sie hierzu bereits Ideen haben, können Sie diese in die Dokumentation der Klasse einfließen lassen.

- (d) Ergänzen Sie die Klasse `Primfaktoren` um eine `main`-Methode, die die Methoden `printPrimfaktoren` und `printGesiebtePrimfaktoren` mit einer Zahl aufruft, die als Parameter über die Kommandozeile (als String) angegeben wird. Um die Ergebnisse aus Teilaufgaben (a) und (b) zu erreichen, kann man dann also die Klasse aufrufen wie folgt:

```
java Primfaktoren 12
```