

## Übungen zu Einführung in die Informatik

### Aufgabe 3-1

### Tauschen

(5./6. 11. 07)

Gegeben sei folgendes Programm:

```
public class Swap{
    public static void main(String[] args){
        int x = 100;
        int y = 10;
        for (int i=0; i<99; i++) {
            // (1)
            x = x + y;
            y = x - y;
            x = x - y;
            // (2)
        }
        System.out.println("x= "+x+" y= "+y); // (*)
    }
}
```

- Was wird ausgegeben, wenn die mit (\*) markierte Zeile erreicht wird?
- Geben Sie für  $i=0, 1, 2, 3, 4$  explizit die Werte der Variablen  $x$  und  $y$  an den mit (1) bzw. (2) markierten Stellen an.

### Aufgabe 3-2

### Größter Gemeinsamer Teiler

(5./6. 11. 07)

In dieser Aufgabe wollen wir eine Methode programmieren, die den größten gemeinsamen Teiler zweier natürlicher Zahlen berechnet und zurückgibt.

- Implementieren Sie eine Klasse `Divisor` mit einer Methode `public static int gcd(int x, int y)`, die den größten gemeinsamen Teiler von  $x$  und  $y$  nach folgendem Algorithmus berechnet:

```
/** Diese Methode gibt den größten gemeinsamen Teiler von den Parametern x und y zurück
 */
public static int gcd(int x, int y){
    Solange x ungleich y ist{
        Falls x < y dann tausche die Werte von x und y;
        x = x - y;
    }
    Gib x zurück;
}
```

- Testen Sie die Methode.
- Terminiert der Algorithmus immer? Warum?
- Wie könnte man ihn verbessern?

### Aufgabe 3-3

### Arithmetisches Mittel

(12./13. 11. 07)

Programmieren Sie eine Klasse `Calculator` mit einer Methode `public static double average(int[] a)`, welche das arithmetische Mittel von einem Integer-Feld (`int[] a`) ermittelt und zurückgibt. Das arithmetische Mittel eines Arrays mit den Elementen  $a_0, a_1, \dots, a_{n-1}$  berechnet sich dabei als  $\frac{1}{n} \sum_{i=0}^{n-1} a_i$ .

### Aufgabe 3-4

### Primzahlen

(12./13. 11. 07)

Primzahlen sind natürliche Zahlen, die nur durch sich und die eins teilbar sind. Die eins ist per definitionem keine Primzahl. Die Primzahlen, welche kleiner als  $n$  sind, kann man u.a. mit dem folgenden Verfahren berechnen:

1. Legen Sie einen `boolean` Array der Länge  $n + 1$  an, und initialisieren die Elemente mit Index 0 und 1 mit `false`. Alle anderen Elemente sollen mit `true` initialisiert werden.
  2. Sei  $x$  der kleinste Index aller Elemente, die den Wert `true` haben (Im ersten Fall also  $x = 2$ ). Setzen Sie alle Elemente, deren Index größer  $x$ , und durch  $x$  teilbar ist, auf `false`.
  3. Wiederholen Sie die Schritte (2) und (3) bis Sie bei  $n$  angekommen sind.
  4. Schreiben Sie abschliessend für jedes Arrayelement mit Wert `true` den entsprechenden Index raus.
- a) Implementieren Sie das Verfahren als `main`-Methode der Klasse `Prime` für die Primzahlen bis 500.
- b) Diskutieren Sie Zeitaufwand und Speicherbedarf, die bei der Berechnung der Primzahlen nach obigem Verfahren entstehen.
- c) Auf welche Weise lässt sich obiger Primzahlalgorithmus optimieren? Wie wirkt sich die Optimierung auf Zeit- und Speicherbedarf aus?

### Aufgabe 3-5

### Wie fängt man einen Elefanten?

(12./13. 11. 07)

Wie fängt man einen Elefanten? Laut <http://jetzt.sueddeutsche.de/texte/anzeigen/297885> gibt es auf die Frage viele Antworten, z.B.:

- Mathematiker jagen Elefanten, indem sie nach Afrika gehen, alles entfernen, was nicht Elefant ist, und ein Element der Restmenge fangen.
- Erfahrene Mathematiker werden zunächst versuchen, die Existenz mindestens eines Elefanten zu beweisen.
- Mathematikprofessoren beweisen die Existenz eines Elefanten und überlassen dann das Aufspüren und Einfangen eines tatsächlichen Elefanten ihren Assistenten.<sup>1</sup>
- Ingenieure jagen Elefanten, indem sie nach Afrika gehen, jedes graue Tier fangen, das ihnen über den Weg läuft, und es einen Elefanten nennen, wenn das Gewicht weniger als 15% von dem eines vorher gefangenen Elefanten abweicht.
- Wirtschaftswissenschaftler jagen keine Elefanten. Aber sie sind fest davon überzeugt, dass Elefanten sich selbst stellen würden, wenn man ihnen nur genug dafür bezahlt.
- Führungskräfte jagen Elefanten, indem sie so lange Meetings abhalten, bis der Elefant sich erlegt hat.
- Microsoft klaut einen Elefanten aus dem Zoo in Seattle, kopiert ihn massenhaft, redet aller Welt ein, dass jeder einen bräuchte, dass dieser die ideale Ergänzung zu MS-Office sei und exportiert 14 Millionen Stück nach Afrika.
- Statistiker jagen das erste Tier, das sie sehen,  $n$  mal und nennen es dann Elefant.
- SQL-Programmierer verwenden folgenden Ausdruck: `SELECT Elefant FROM Afrika`.

Studenten dieser Vorlesung programmieren eine Klasse `Hunting` mit den folgenden Methoden:

```
public static int countElephants(boolean[] [] africa), welche die Anzahl aller Elefanten im zwei-dimensionalen Array africa zurückgibt, wobei ein Feld den Wert true hat, falls sich auf diesem Feld ein Elefant befindet, sonst false;
```

---

<sup>1</sup>Leicht modifiziert vom Original.

`public static int[] findAnElephant(boolean[][] africa)`, welche die x- und y-Koordinaten eines Feldes, auf dem sich ein Elefant befindet, in einem Array zurückgibt. Der zurückgegebene Array soll zwei Elemente enthalten. Das erste speichert den x-Koordinaten, das zweite den y-Koordinaten.

### Aufgabe 3-6

### Eingabeparameter

(12./13. 11. 07)

Die ausgezeichnete Methode `public static void main(String[] args)` dient einem Java-Programm als Einstiegspunkt. Sie wird beim Programmstart von der Laufzeitumgebung in der angegebenen Hauptklasse aufgerufen. Zusätzliche Parameter, welche beim Programmstart in der Kommandozeile eingegeben werden, werden im Array `String[] args` abgelegt und können in der `main`-Methode vom Programmierer verwendet werden.

Beispiel: Wir starten eine Klasse `Test` mit dem Aufruf `java Test ABC 123 A1B2` in der Kommandozeile. Im Array `args` der `main`-Methode sind dann die folgenden Werte abgespeichert: `{"ABC", "123", "A1B2"}`.

Programmieren Sie eine Klasse `InputParameters` mit einer `main`-Methode, die folgendes leistet:

```
public static void main(String[] args)
{
    // Schreiben Sie alle Elemente im Array args auf den Bildschirm
    // Beispiel: nach dem Aufruf
    // java InputParameters ABC 123 A1B2
    // soll Ihr Programm folgendes ausgeben:
    //
    // Element 1: ABC
    // Element 2: 123
    // Element 3: A1B2

    // Schreiben Sie alle Elemente im Array args in umgekehrter Reihenfolge
    // auf den Bildschirm.
    // Die einzelnen Elemente sollen ebenfalls in umgekehrter Buchstabenreihenfolge
    // auf dem Bildschirm erscheinen. Verwenden Sie zum Umkehren der Buchstaben-
    // reihenfolge ein char[], das sie vorher mit Hilfe der Methode charAt(int index)
    // der Klasse String initialisiert haben.
    //
    // Beispiel: nach dem Aufruf
    // java InputParameters ABC 123 A1B2 soll jetzt folgendes ausgegeben
    // werden:
    //
    // Element 3: 2B1A
    // Element 2: 321
    // Element 1: CBA
}
```