

### Aufgabenblatt 3 *Programmieren I*

#### Hinweise

- Die Abgabe dieser Übungsaufgaben muss bis spätestens Sonntag, den 14. November 2021 um 23:59 Uhr im ISIS-Kurs erfolgt sein.
- Lösungen zu diesen Aufgaben sind als gezippter Projektordner abzugeben. Eine Anleitung zum Zippen von Projekten finden Sie auf der Seite des ISIS-Kurses.  
*Bitte benutzen Sie einen Dateinamen der Form VornameNachname.zip.*

#### Aufgabe 3.1 *Inhomogene Arrays, Klassenmethoden und Schleifen (4 Punkte)*

Erstellen Sie ein neues Projekt **Aufgabe3** und darin eine neue Klasse mit Namen **InhomogeneousArrays**. Ein inhomogenes Array in zwei Dimensionen hat „Zeilen“ verschiedener Länge. Folgendes ist ein Beispiel einer inhomogenen Array-Konstante vom Typ `char[][]`:

```
{
    {'t','o','b','i','a','s'},
    {'h','e','i','n','d','e','l'},
    {'1','2','1','6','3'}
}
```

Diese Klasse **InhomogeneousArrays** soll, neben einer **main**-Methode, die Klassenmethoden **makeIntoOneLine**, **mirror**, **swapRows**, und **upsideDown** haben.

- Die Methode **makeIntoOneLine** hat einen Formalparameter, und zwar ein inhomogenes, zweidimensionales **char**-Array, und gibt einen **String**-Wert zurück, in dem die „Zeilen“ durch Leerzeichen getrennt nacheinander stehen. Die Rückgabe für obiges Array wäre also folgender **String**:

```
"tobias heindel 12163"
```

- Die Methode **mirror** hat einen Formalparameter vom Typ `char[][]` und kehrt die Reihenfolge der **char**-Werte in jeder „Zeile“ des Arrays um. Die Änderung für obiges Beispiel ist folgende:

```
{
    {'s','a','i','b','o','t'},
    {'l','e','d','n','i','e','h'},
    {'3','6','1','2','1'}
}
```

**Tipp** Wenn `x` vom Typ `char[][]` ist, dann ist `x.length` die Anzahl der „Zeilen“ und `x[i].length` die Länge der `i`-ten „Zeile“.

- Die Methode `swapRows` hat einen Formalparameter vom Typ `char[][]` und zwei Formalparameter vom Typ `int` und die Methode vertauscht die beiden „Zeilen“, die den beiden `int`-Werten entsprechen. Vertauschen der „Zeile“ 0 mit der „Zeile“ 1 des obigen Beispiellarrays liefert das Folgende:

```
{
    {'h','e','i','n','d','e','l'},
    {'t','o','b','i','a','s'},
    {'1','2','1','6','3'}
}
```

- Die Methode `upsideDown` hat einen Formalparameter vom Typ `char[][]` und kehrt die Reihenfolge der „Zeilen“ um. Für das obige Beispiel wäre der Effekt wie folgt:

```
{
    {'1','2','1','6','3'},
    {'h','e','i','n','d','e','l'},
    {'t','o','b','i','a','s'}
}
```

- Schließlich, in der `main`-Methode,
  - definieren Sie eine Variable vom Typ `char[][]` und weisen Sie dieser eine Konstante mit drei „Zeilen“ zu, die jeweils Ihrem Vornamen, Ihrem Nachnamen und Ihrer Postleitzahl entsprechen,
  - rufen Sie die Methode `makeIntoOneLine` mit der Variable auf und geben Sie das Ergebnis auf der Konsole aus,
  - rufen Sie die Methode `mirror` mit der Variable auf,
  - rufen Sie die Methode `makeIntoOneLine` mit der Variable auf und geben Sie das Ergebnis auf der Konsole aus,
  - nutzen Sie die Methode `swapRows`, um die ersten beiden „Zeilen“ der Variable zu vertauschen,
  - rufen Sie die Methode `makeIntoOneLine` mit der Variable auf und geben Sie das Ergebnis auf der Konsole aus,
  - rufen Sie die Methode `upsideDown` mit der Variable auf,
  - rufen Sie die Methode `makeIntoOneLine` mit der Variable auf und geben Sie das Ergebnis auf der Konsole aus.

Ein Beispielablauf würde wie folgende Ausgabe liefern.

```
tobias heindel 12163
saibot lednieh 36121
lednieh saibot 36121
36121 saibot lednieh
```

Testen Sie Ihr Programm.

### **Aufgabe 3.2**    *Langtons Ameise (3 Punkte)*

In dieser Aufgabe geht es darum, Langton's Ameise zu implementieren; das Vorgehen der Ameise ist wie folgt beschrieben (basierend auf diesem Wikipedia-Artikel [https://de.wikipedia.org/wiki/Ameise\\_\(Turingmaschine\)#Der\\_Algorithmus](https://de.wikipedia.org/wiki/Ameise_(Turingmaschine)#Der_Algorithmus)).

Die Ameise bewegt sich in einem rechteckigen Array von `char`-Werten, die entweder `' '` oder `'X'` sein können. In der Ausgangssituation befindet sich die „Ameise“ in der Mitte des Arrays und schaut in eine bestimmte Richtung (0, 90, 180 oder 270 Grad). Der Übergang zum nächsten Zustand erfolgt nach folgenden Regeln:

- (a) Auf einer `' '`-Zelle drehe 90 Grad nach rechts; auf einer `'X'`-Zelle drehe 90 Grad nach links.
- (b) Wechsle den Typ des Feldes (`' '` wird `'X'` und umgekehrt—also `'X'` wird `' '`).
- (c) Gehe zur nächsten Zelle in der aktualisierten Richtung ...

... es sei denn, die „Ameise“ würde das Array verlassen; am Anfang wollen wir den mittleren Teil des Arrays mit zufälligen Werten initialisieren (und den Rest mit Leerzeichen). Im Detail gehen Sie bitte wie folgt vor. Erstellen Sie eine neue Klasse mit Namen `LangtonsAnt`. In der `main`-Methode,

- (a) erstellen Sie ein rechteckiges Array der Größe  $100 \times 100$  von Zeichen,
- (b) initialisieren Sie alle Zellen mit dem Leerzeichen `' '` mit der Ausnahme, dass Sie
- (c) ein Quadrat von  $100 = 10 \times 10$  Zellen in der Mitte zufällig mit `'X'` oder `' '` belegen,
- (d) lassen Sie dann die „Ameise“ in der Mitte des Arrays los und lassen Sie sie laufen bis sie das Array verlassen würde und
- (e) am Schluss geben Sie das Array auf der Konsole aus indem Sie die „Zeilen“ des Arrays nacheinander ausgeben.

Testen Sie Ihr Programm. Eine mögliche Ausgabe finden Sie in Abbildung 1.

### **Aufgabe 3.3**    *Das Pascalsche Dreieck per Rekursion (1 Zusatzpunkt)*

Erstellen Sie eine neue Klasse mit Namen `PascalsTriangle`. In der `main`-Methode

- lesen Sie eine natürliche Zahl  $n$  ein und
- geben Sie die ersten  $n$  Zeilen des Pascalschen Dreiecks aus, und zwar
  - **ohne** Verwendung von Schleifen.

Natürlich dürfen Sie Klassenmethoden verwenden. Folgendes ist eine mögliche Ausgabe.

Bitte eine natürliche Zahl größer 0 eingeben:5

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

Abbildung 1: Langtons Ameise