

# Aufgabenblatt 2

## Kompetenzstufe 1

### Allgemeine Informationen zum Aufgabenblatt:

- Die Abgabe erfolgt in TUWEL. Bitte laden Sie Ihr IntelliJ-Projekt bis spätestens **Freitag, 16.11.2018 13:00 Uhr** in TUWEL hoch.
- Zusätzlich müssen Sie in TUWEL ankreuzen, welche Aufgaben Sie gelöst haben und während der Übung präsentieren können.
- Ihre Programme müssen kompilierbar und ausführbar sein.
- Ändern Sie bitte nicht die Dateinamen und die vorhandene Ordnerstruktur.
- Bei manchen Aufgaben finden Sie Zusatzfragen. Diese Zusatzfragen beziehen sich thematisch auf das erstellte Programm. Sie müssen diese Zusatzfragen für gekreuzte Aufgaben in der Übung beantworten können. Sie können die Antworten dazu als Java-Kommentare in die Dateien schreiben.
- Verwenden Sie, falls nicht anders angegeben, für alle Ausgaben `System.out.println()` bzw. `System.out.print()`.
- Verwenden Sie für die Lösung der Aufgaben keine Aufrufe (Klassen) aus der Java-API, außer diese sind ausdrücklich erlaubt, wie z.B. die Klassen `StdDraw` und `Scanner` oder Klassen, die in den Hinweisen zu den einzelnen Aufgaben aufscheinen.

### In diesem Aufgabenblatt werden folgende Themen behandelt:

- for-Schleife, while-Schleife
- Verschachtelung von Schleifen
- Zeichnen mit `StdDraw` unter Verwendung von Schleifen und Verzweigungen

# Aufgabe 1

Erweitern Sie für Teilaufgabe A die Methode `main`:

- Schreiben Sie ein Programm, das bei einer gegebenen ganzen Zahl `int maxNumber` alle *Harshad-Zahlen* (von 1 bis einschließlich `int maxNumber`) berechnet. Es handelt sich dann um eine Harshad-Zahl (HZ), wenn die Zahl durch die Quersumme aller Ziffern der Zahl mit 0 Rest teilbar ist.

Beispiele:

$1 \rightarrow 1 \rightarrow 1 \% 1 = 0 \rightarrow$  teilbar und somit eine HZ

$97 \rightarrow 9 + 7 = 16 \rightarrow 97 \% 16 = 1 \rightarrow$  nicht teilbar und somit keine HZ

$777 \rightarrow 7 + 7 + 7 = 21 \rightarrow 777 \% 21 = 0 \rightarrow$  teilbar und somit eine HZ

$8316 \rightarrow 8 + 3 + 1 + 6 = 18 \rightarrow 8316 \% 18 = 0 \rightarrow$  teilbar und somit eine HZ

$9214 \rightarrow 9 + 2 + 1 + 4 = 16 \rightarrow 9214 \% 16 = 14 \rightarrow$  nicht teilbar und somit keine HZ

Geben Sie alle Harshad-Zahlen von 1 bis einschließlich `maxNumber = 1000` mittels `System.out.println()` auf der Konsole aus.

- ❗ Die Teilaufgabe A soll ohne die Verwendung von Strings und Arrays gelöst werden.

**Zusatzfrage(n) zu Teilaufgabe A:**

1. Welche Art von Schleife(n) verwenden Sie für die Lösung dieses Beispiels? Begründen Sie Ihre Wahl.

Erweitern Sie für Teilaufgabe B die Methode `main`:

- Durchlaufen Sie alle Zahlen von 1 bis einschließlich 500 und konkatenieren Sie alle Zahlen, die durch 15 teilbar sind in einer String-Variablen. Verwenden Sie das Zeichen '#' als Trennzeichen, damit die Zahlen in der Form `#15#30#...#480#495#` auf der Konsole angezeigt werden.
- Nun ersetzen Sie in der erstellten String-Variablen alle Vorkommen von '1' durch das Zeichen '!' und alle Vorkommen von '5' durch das Zeichen '\$'. Geben Sie den geänderten String erneut auf der Konsole aus.

- ❗ Bei Teilaufgabe B soll für die Manipulation des Strings keine der `replace`-Methoden verwendet werden. Auch Arrays sind hier nicht erlaubt!

**Zusatzfrage(n) zu Teilaufgabe B:**

1. Wie müssen Sie Ihr Programm ändern, damit Sie alle Vorkommen von '2' durch die Zeichen "\*\*\*" ersetzen können?
2. Wie müssen Sie Ihr Programm ändern, damit Sie alle Vorkommen von "10" durch das Zeichen '+' ersetzen können?

## Aufgabe 2

Erweitern Sie die Methode `main`:

- Implementieren Sie ein Programm, welches die unten gezeigten Rautenfiguren erzeugt. Die Höhe der Figuren wird mit der Variablen `int figHeight` gesteuert. Die Höhe gibt die Anzahl der Zeilen an und darf nur positive ungerade Werte annehmen. Sie können den Wert für `figHeight` gerne mittels Scanner einlesen.

Beispiele:

```

+++++++\++++++
++++++/00\++++++
++++++/0000\++++++
++++++/000000\++++++
++++++/00000000\++++++
+++\\0000000000\\+++
++\\000000000000\\++
+/000000000000000\\+
|0000000000000000|
+\\000000000000000/+
++\\000000000000/+
+++\\0000000000/+
++++\\00000000/+
+++++\\00000/+
+++++\\0000/+
+++++\\00/+
+++++\\/+

+++++\\+++++
+++++\\00\\+++++
+++\\0000\\+++
++\\000000\\++
+/00000000\\+
|0000000000|
++\\00\\+++
+/0000\\+
+++\\++++
++\\00\\++
++\\0000\\+
++\\00000|
++\\0000/+
++\\0000/+
++\\00/+
+++\\/+

```

Ausgaben (von links nach rechts) mit `figureHeight` 17, 11, 7 und 1.

- ❗ Die Zeichen innerhalb der Raute entsprechen dem Buchstaben 'O'. Damit der Backslash ausgegeben werden kann, müssen Sie '\\\\' schreiben. Es handelt sich dabei um eine sogenannte *Escape Sequence*.
- ❗ Für die Realisierung des Beispiels dürfen keinerlei Strings oder Arrays verwendet werden. Auch Methoden aus diesen Klassen dürfen nicht zum Einsatz kommen.

Zusatzfrage(n):

1. Kann man das Beispiel mit einer einzelnen Schleife lösen? Wenn ja, wie und an welchen Stellen müssten Sie Codeänderungen durchführen?

## Aufgabe 3

Erweitern Sie die Methode `main`:

- Implementieren Sie die in Abbildung 1 gezeigte optische Täuschung. Die optische Täuschung besteht aus 10 Quadraten horizontal und 11 Quadraten vertikal. Die Quadrate haben eine Abmessung von  $40 \times 40$  Pixel.
- Verwenden Sie für die Quadrate folgende Farben:
  - für dunkelgrüne Quadrate `StdDraw.setPenColor(90, 90, 0)`
  - für hellgrüne Quadrate `StdDraw.setPenColor(166, 166, 76)`.
- Zeichnen Sie bei jedem Kreuzungspunkt zwei übereinander liegende Rechtecke (siehe Abbildung) ein. Die Länge der Rechtecke beträgt 20 Pixel und die Breite 4 Pixel. Achten Sie dabei darauf, wie die schwarzen und weißen Rechtecke angeordnet werden.
- Verwenden Sie für die Implementierung des kompletten Programms zwei ineinander verschachtelte for-Schleifen.

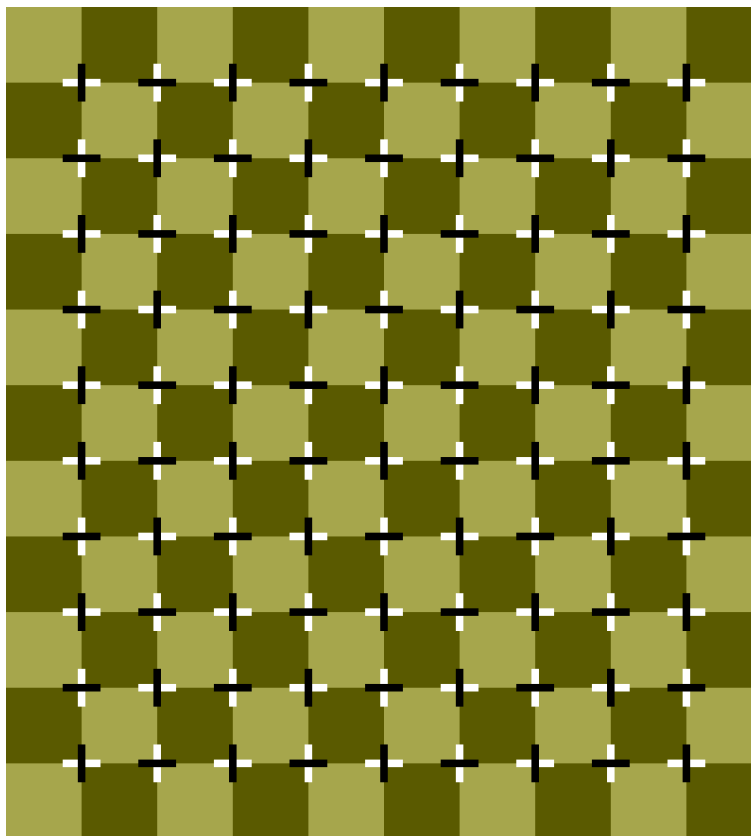


Abbildung 1: Optische Täuschung mit Quadraten und den schwarzweißen Kreuzen

**Zusatzfrage(n):**

1. Wie müssen Sie den Code schreiben, damit Sie nur durch Änderung der drei Variablen für horizontale Quadrate, für vertikale Quadrate und für die Abmessung der Quadrate ein neues Ergebnis bekommen?