

# Einführung in die Informatik

Institut für Eingebettete Systeme/Echtzeitsysteme | Wintersemester 2020/21

Valentina Richthammer, Michael Gläß

## Übungsblatt 5: Arrays

Abgabetermin: 13.12.2020, 23:59 Uhr

Geben Sie **Programmieraufgaben als Java Code (\*.java Dateien)** ab. Alle anderen Aufgaben, die Text oder Grafiken erfordern, geben Sie **als PDF Dateien** ab. PDFs können Sie beispielsweise mit dem kostenlosen Programm *LibreOffice* erstellen. Alternativ können Sie etwas mehr Zeit investieren und LaTeX lernen, was Sie im späteren Studium immer wieder brauchen werden.

Wenn Sie **mehrere Dateien** abgeben wollen, dann fassen Sie diese zu **einem ZIP File** zusammen.

Präsenzaufgaben werden direkt im Tutorium bearbeitet, werden nicht bepunktet und müssen nicht abgegeben werden.

### Präsenzaufgabe

Programmieren Sie in Java einen Algorithmus, der die Werte eines Integerarrays in umgekehrter Reihenfolge in ein neues Array speichert.

### Aufgabe 1: Verschachteln

(5)

Schreiben Sie ein Javaprogramm, das für alle Dreier-Kombinationen von Einträgen eines Integerarrays deren Summe berechnet. Alle Dreier-Kombinationen sowie deren Summe sollen auf der Konsole ausgegeben werden.

### Aufgabe 2: Matrizenmultiplikation

(8)

Programmieren Sie einen Algorithmus, der eine Matrizenmultiplikation durchführen kann. Die beiden Matrizen und auch die Ergebnismatrix werden durch zweidimensionale Integerarrays dargestellt. Sie können davon ausgehen, dass die Arraydimensionen  $> 0$  sind und die Arrays valide Matrizen abbilden. Allerdings wissen Sie nicht, ob die Matrizen für die Multiplikation passende Dimensionen besitzen.

Geben Sie außerdem die Ergebnismatrix auf der Konsole aus.

Ein anschauliches Beispiel einer Matrizenmultiplikation finden Sie unter:

<https://de.wikipedia.org/wiki/Matrizenmultiplikation>.

### Zusatzaufgabe (2/6): n Türme

(6)

Die Punkte der Zusatzaufgaben zählen *nicht* zum aktuellen Übungsblatt! Wenn Sie mind. die Hälfte aller 6 Zusatzaufgaben bestehen, können Sie damit ein nicht-bestandenes Übungsblatt ausgleichen. Eine Zusatzaufgabe ist bestanden, wenn mind. 50% der möglichen Punkte erreicht wurden.

In dieser Aufgabe beschäftigen wir uns mit dem  $n$ -Türme Problem:

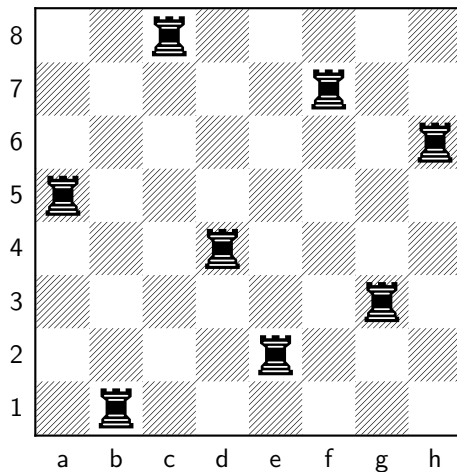
Auf einem  $n \times n$  Felder großen Spielfeld sollen  $n$  Türme so positioniert werden, dass sie sich nicht gegenseitig schlagen

können ( $n > 0$ ).

Türme können gerade nach oben, unten, rechts oder links ziehen. Dabei können sie beliebig viele Felder weit ziehen, aber nur in eine Richtung.

Ein Turm schlägt einen anderen Turm, wenn er auf dessen Feld zieht.

Folgende Abbildung zeigt eine korrekte Lösung für das  $n$ -Türme Problem ( $n = 8$ ) :



Das Spielfeld wird mit einem  $n \times n$  großen boolean-Array dargestellt, wobei jeder Eintrag des Arrays ein Feld repräsentiert. Ein Eintrag ist `true` wenn auf dem entsprechenden Feld ein Turm positioniert ist.

Schreiben Sie ein Programm, das überprüft ob  $n$  Türme korrekt aufgestellt wurden. Das Programm soll dann entsprechend auf der Konsole ausgeben, ob die Positionierung korrekt oder falsch ist.