

## **Präsenzaufgaben 9**

**27./28.05.2024**

Die Lösung der Aufgaben wird am Ende der Übung von Ihnen vorgestellt.

### **Aufgabe 1**

Schreiben Sie eine Klasse `Permutationen`, in der Sie die Lösungen der Aufgaben 1-3 einfügen.

Schreiben Sie eine Funktion

```
public static void printPermutationen(int n)
```

die alle Permutationen der Zahlenfolge  $1,2,\dots,n$  auf dem Bildschirm ausgibt.

Beispiel: Die möglichen Permutationen der Zahlenfolge 1,2,3 sind:

```
1,2,3
1,3,2
2,1,3
2,3,1
3,1,2
3,2,1
```

$n$  kann auch größer als 9 sein. Als Trennzeichen können beliebige Zeichen verwendet werden.

### **Aufgabe 2:**

Schreiben Sie eine Funktion

```
public static void printPermutationen2(int n, int ges)
```

die alle Strings auf dem Bildschirm ausgibt, die die folgenden Eigenschaften erfüllen:

- Der String besteht ausschließlich aus den Zeichen 0 und 1.
- Die Länge des Strings ist  $ges$ .
- Der String enthält genau  $n$  Einsen.

Benutzen Sie einen rekursiven Backtracking-Algorithmus

Beispiel: Der Aufruf `printPermutationen2(2, 4)` gibt folgende Strings auf dem Bildschirm aus:

```
1100, 1010, 1001, 0110, 0101, 0011
```

Die Reihenfolge ist dabei nicht maßgeblich.

**Aufgabe 3:**

Schreiben Sie eine Funktion

```
public static void printPermutationen3(int n, int ges)
```

Lösen Sie Aufgabe 2 noch einmal, aber verwenden Sie diesmal statt der Rekursion eine Queue und einen Level-Order-Durchlauf.

**Aufgabe 4:**

Betrachten Sie die Aufgabe

<https://www.geeksforgeeks.org/remove-invalid-parentheses/>

und die Java-Lösung dazu.

Das Problem hat eine Baumstruktur, die durchlaufen wird. Zeichnen Sie die beiden Bäume, die zu den Beispielen "( ) ( ) ( )" und "( ) ∇)" gehören.