

Minimumsuche

1. Pseudocode

Schreibe in *Pseudocode* (d.h. in „freier Sprache“), wie die Minumumsuche funktioniert, also wie in einem gegebenen, unsortierten Array, der kleinste Wert herausgefunden werden kann:

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins or other markings on the paper.

Einführung Zufallszahlen

Es bietet sich an, nicht mit einem vordefinierten Array zu arbeiten, sondern die Zahlen im Array *zufällig* zu erzeugen.

Java bietet hierfür die Möglichkeit Zufallszahlen¹ zu erzeugen.

Für eine bessere Übersichtlichkeit rechnen wir hier vorerst nur mit ganzen Zahlen. Mit dem Befehl `Math.random()`; erzeugen wir jedoch lediglich Fließkommazahlen zwischen 0 und 1.

Für ganzzahlige Zufallszahlen müssen wir zunächst das Paket `java.util.Random` importieren. Anschließend können wir in unserem Programmablauf einen Zufallszahlengenerator erzeugen:

¹Anmerkung: es können mit einem herkömmlichen Computer keine *echten Zufallszahlen* erzeugt werden, diese werden mit einem aufwändigen Algorithmus *berechnet*. Je besser dieser Algorithmus ist, desto zufälliger sehen die Zahlen aus. Wir sprechen deshalb auch von *Pseudozufallszahlen*.

```
import java.util.Random;

class Zufallszahl {
    public static void main(String[] args) {
        // erzeuge Zufallszahlengenerator
        Random rand = new Random();

        //erzeuge Zufallszahl zwischen >=0 und < 50
        rand.nextInt(50);
    }
}
```

Listing 1: Erzeugung von Zufallszahlen

Der Parameter der `rand.nextInt`-Methode gibt dabei die obere Grenze der Zufallszahlen an. Im obigen Beispiel liegen die erzeugten Zufallszahlen also immer zwischen *inklusive* 0 und *exklusive* 50.

2. Zufallszahlengenerator

Erzeuge ein neues Java-Projekt **Sortierung**. An diesem Projekt werden wir die nächsten Wochen arbeiten.

Lege darin ein Paket **minimumsuche** mit einer Klasse **Minimum** (inklusive `main`-Methode) an. Programme hier zunächst einen Zufallszahlengenerator:

- Erzeuge zunächst ein Array, welches 20 Ganzzahlen speichern kann.
- Befülle dieses Array mit 20 zufälligen Zahlen (zwischen 0 und 50).
- Lasse die Werte dieses Arrays auf der Konsole kommagetrennt ausgeben.
Beispiel: `20,6,30,34,5,11,0,34,28,12,4,26,11,15,44,28,40,7,20,7`

3. Minimumsuche

Erweitere den Programmablauf aus Aufgabe 2 so, dass im zufällig befüllten Array der Minimale Wert und der zugehörige Index gesucht und ausgegeben wird. Verwende dazu den Ablauf aus Aufgabe 1.

Beispielausgabe: `Index: 6, Wert: 0`

4. Zusatzaufgabe: Minimumsuche als Methode

Um die Minimumsuche nicht jedes Mal erneut programmieren zu müssen soll nun eine passende Methode programmiert werden. Das zu durchsuchende Array soll dabei als *Parameter* an die Methode übergeben werden.

Überlege dir zunächst, was als Ergebnis der Methode zurückgegeben werden soll und programmiere anschließend diese Methode in die **Minimum**-Klasse.

5. Zusatzaufgabe 2: Sortierte Ausgabe

Überlege dir, wie man diese Minimumssuche dazu verwenden könnte, alle Einträge des Arrays sortiert auf der Konsole auszugeben.