Programmierung: Java 2

Jürgen Hermes – IDH – SoSe 2025

Java Collections Framework – Themen

- Konzepte: Iterator, Iterable, Generics
- Lists – Konzepte und Funktionalität
  - Implementation: ArrayList
  - Implementation: LinkedList
- Sets – Konzepte und Funktionalität
  - Exkurs: Objekt-Gleichheit
  - Implementation: HashSet
- Maps – Konzepte und Funktionalität
  - **Implementation: HashMap**
  - **Exkurs: Rekursion**
  - **Implementation: TreeMap**

Rekursion

Rekursion

- Als Rekursion (lateinisch recurrere ‚zurücklaufen‘) wird ein prinzipiell unendlicher Vorgang bezeichnet, der sich selbst als Teil enthält oder mithilfe von sich selbst definierbar ist. (Wikipedia)

Rekursive Algorithmen (rufen sich selbst auf)

- Algorithmen, die sich selbst aufrufen, um ein Teilproblem zu lösen.
- Beispiel: Rekursive Definition natürlicher Zahlen
  - 0 ist eine natürliche Zahl UND
  - Eine natürliche Zahl + 1 ergibt wieder eine natürliche Zahl.
- Beispiel: Fakultät
  - Nicht-Rekursiv: $n! := \prod_{k=1}^{n} k = 1 \cdot 2 \cdot 3 \cdots n$
  - Rekursiv: 0! = 1 UND n! = n * (n-1)!
- Rekursive Algorithmen benötigen immer zwei Komponenten:
  - Basisfall (Abbruchbedingung): Wann und wie endet die Rekursion?
  - Rekursiver Schritt: Ruft sich selbst wieder auf.

Rekursive Datenstrukturen (enthalten sich selbst)

- Datenstrukturen, bei denen ein Element (z. B. ein Objekt oder eine Klasse) Referenzen auf Instanzen derselben Datenstruktur enthalten.
- Beispiel: `class ListNode {int value; ListNode next;}`
- Weiteres Beispiel: Bäume
  - Jeder Baum hat 0 oder 1 Elternteil und 0 oder mehr Kinder.
  - Wurzel (Root): Kein Elternteil; Blatt (Leaf): Keine Kinder
  - Verzeichnisbäume: class Directory{String name; List<Directory> subDirectories;}
  - Binärbäume: class BinaryTreeNode {int value; BinaryTreeNode left; BinaryTreeNode right;}
  - Suchbäume: Elemente sortiert einfügen zur schnellen Auffindbarkeit.