



Objektorientierte Programmierung

ArrayList Einführung

Prof. Dr. Ulrike Hammerschall
Fakultät für Informatik und Mathematik

Das Collection Framework in Java



- Sammlung generischer Interfaces und Klassen zur Verwaltung großer Datenmengen.
- Generische Klassen: Der Typ der Objekte, die gespeichert werden dürfen, wird bei der Initialisierung festgelegt.
- Ziel ist die dynamische Verwaltung großer Datenmengen in einem Programm:
 - Effizienter Zugriff auf Daten: Lesen, Schreiben, Suchen, Ändern, ...
 - Effiziente Algorithmen auf Daten: Vergleichen, Sortieren, Maximum, Minimum, Füllen, Umdrehen, Rotieren, Durcheinanderwürfeln, ...
- Zwei grundsätzliche Speicherstrukturen:
 - Collections: Speicherung von Einzelelementen
 - Listen (List): **ArrayList**, LinkedList
 - Mengen (Set): HashSet, TreeSet
 - Assoziative Speicher: Speicherung von Elementpaaren
 - Maps (Map): HashMap, TreeMap

ArrayList als Container-Datenstruktur



- Modernisierte Variante des relativ unflexiblen und einfachen Arrays.
- Verwaltet Elemente intern in einem Array.
- Passt dynamisch seine Länge an die Anzahl der Elemente an.
- Verwaltung der Elemente über Index mit Start bei 0.
- Alle Zugriffe auf die Listenelemente erfolgen über Methoden.
- Als Elementtypen sind **ausschließlich** Referenztypen zulässig.
- Für Basisdatentypen stehen entsprechende Wrapper-Klassen zur Verfügung (Integer, Double, Boolean, ...).

Methodenauswahl 1



Methode	Beschreibung
add(E e): boolean	Stellt sicher, dass das Element e in der Collection enthalten ist. Liefert true, falls die Collection geändert wurde, sonst false. Muss in den konkreten Implementierungen entsprechend redefiniert werden.
clear()	Entfernt alle Elemente aus der Collection.
contains(Object o): boolean	Liefert true, falls die Collection ein inhaltlich gleiches Objekt o enthält.
remove(Object o): boolean	Entfernt maximal eine Instanz des gesuchten Objekts. Liefert true, falls ein Objekt gelöscht wurde.
size(): int	Liefert die Anzahl der Elemente in der Collection.
toArray(): Object[]	Liefert ein Array, welches die Elemente der Collection enthält.
isEmpty(): boolean	Liefert true, falls die Collection keine Elemente enthält

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Collection.html>

Methodenauswahl 2



Methode	Beschreibung
<code>add(int index, E e)</code>	Fügt das Element <code>e</code> an der Stelle <code>index</code> in der Liste ein. Alle nachfolgenden Elemente werden um eine Stelle nach hinten verschoben.
<code>get(int index): E</code>	Liefert das Element an der Stelle <code>index</code> . Die Liste wird nicht verändert.
<code>set(int index, E e): E</code>	Ersetzt das Element an der Stelle <code>index</code> durch das Element <code>e</code> . Liefert das ursprüngliche Element zurück.
<code>indexOf(E e): int</code>	Liefert den Index des ersten Vorkommens des Elements <code>e</code> in der Liste oder <code>-1</code> , falls das Element nicht gefunden wurde.
<code>lastIndexOf(E e): int</code>	Liefert den Index des letzten Vorkommens des Elements <code>e</code> in der Liste oder <code>-1</code> , falls das Element nicht gefunden wurde.
<code>remove(int index)</code>	Entfernt das Element an der Stelle <code>index</code> .

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/List.html>

Anwendung einer ArrayList (siehe auch Moodle)



```

public class ArrayListExample {
    // Objektvariable
    private ArrayList<String> list;

    // Konstruktor
    ArrayListExample() {
        list = new ArrayList<>();
    }

    void addElementToList(String element) {
        list.add(element);
    }

    void addElementToList(int index, String element) {
        list.add(index, element);
    }

    String getElementFromList(int index) {
        return list.get(index);
    }

    void removeElementFromList(int index) {
        list.remove(index);
    }

    void removeElementFromList(String element) {
        // nur möglich bei Identität oder falls die Methoden
        // equals() und hashCode() implementiert sind.
        list.remove(element);
    }

    void printList() {
        System.out.println(list);
    }

    // main-Methode zum Start
    public static void main(String[] args) {
        // Beispielmethode zur Anwendung der Klasse
        ArrayListExample example = new ArrayListExample();

        example.addElementToList("Hello");
        example.printList();

        String value = example.getElementFromList(0);
        example.removeElementFromList("Hello");
        example.printList();

        ArrayList<String> list = new ArrayList<>(List.of("eins", "zwei"));
        list.add(0, "drei");
        System.out.println(list);

        // for-Schleife über Elemente von Container-Klassen
        for (String next : list) {
            System.out.println(next);
        }
    }
}

```