

Übung

Collections

Aufgabe 1

Geben Sie den Code für die Erstellung einer **ArrayList** an, die nur

- a) String-Elemente aufnehmen kann,
- b) int-Elemente aufnehmen kann,
- c) char-Elemente aufnehmen kann,
- d) Objekte einer bereits vorhandenen Klasse `Mitarbeiter` aufnehmen kann.

Aufgabe 2

Beantworten Sie die folgenden Fragen zu **ArrayLists** schriftlich. Schreiben Sie *danach* jeweils ein kurzes Programm, mit dem die Antwort ermittelt werden kann.

- a) Was passiert beim Einfügen eines Elements in eine `ArrayList` mit den nachfolgenden Elementen?
- b) Was passiert beim Löschen eines Elements in einer `ArrayList` mit den nachfolgenden Elementen?
- c) Kann zu einer `ArrayList` mit 10 Elementen ein weiteres Element an Index 20 hinzugefügt werden?

Aufgabe 3

Es soll ein Programm zum Auswerten von Messwerten (Datentyp `double`) mit Hilfe einer **ArrayList** erstellt werden.

- Erstellen Sie ein Konsolenprogramm, das eine `ArrayList` für `double`-Werte erstellt.
- Fügen Sie der `ArrayList` folgende Elemente hinzu: 4.3, -9, 0, 11.12, 99.
- Lassen Sie sich alle Elemente mit Hilfe einer `for-each`-Schleife ausgeben.

- Fügen Sie an Index 3 den Messwert 0.25 hinzu. Wie wird die Liste jetzt aussehen? Schreiben Sie die Elemente der Liste in der richtigen Reihenfolge auf, *bevor* Sie Ihre Antwort überprüfen, indem Sie die Liste erneut mit einer for-each-Schleife ausgeben lassen
- Entfernen Sie das Element an Index 1. Wie wird die Liste jetzt aussehen? Schreiben Sie *zuerst* die Elemente der Liste in der richtigen Reihenfolge auf. *Danach* überprüfen Sie Ihre Antwort, indem Sie die Liste erneut mit einer for-each-Schleife ausgeben lassen.

Aufgabe 4

Schreiben Sie ein Programm, dass feststellt, ob in einem eingegebenen Text alle Klammern in korrekten Paaren auftreten.

Beispiele

- `()` OK
- `()` NOK
- `()()` NOK
- `((()()))` OK
- `((()()))` NOK
- `((()((()))))` NOK

Hinweise

- Verwandeln Sie den eingegebenen String zunächst in ein char-Array. Dazu stellt die Klasse String die Methode `toCharArray` zur Verfügung.
- Es reicht nicht zu zählen, ob gleich viele öffnende wie schließende Klammern vorkommen, da auch die Reihenfolge wichtig ist. Verwenden Sie stattdessen einen **Stack**.

Aufgabe 5

- a) Schreiben Sie ein Programm, das zehn Postleitzahlen einliest und am Ende alle eingegebenen Postleitzahlen ohne Duplikate ausgibt. Benutzen Sie die Datenstruktur **HashSet**.
- b) Schreiben Sie ein Programm, das zwei Folgen von jeweils fünf Postleitzahlen einliest und am Ende nur die Postleitzahlen ausgibt, die in beiden Folgen vorkamen. Benutzen Sie wieder die Datenstruktur **HashSet**.