

## Quizzies

Aktuelle Punktzahl: 0 / 13

Beantwortet: 0 / 14

Eine Sammlung kann andere Objekte speichern und verwalten. Sammlungen bieten typischerweise Möglichkeiten an, auf diese Objekte zuzugreifen.

- ☐ Trifft nicht zu
- ☒ Trifft zu

Antwort speichern

➔ 1 Punkt

## Quizzies

Aktuelle Punktzahl: 1 / 13

Beantwortet: 1 / 14

Was trifft zu?

- ☒ Das folgende führt zu einem Kompilierfehler:  

```
ArrayList<String> liste1 = new ArrayList<String>();  
ArrayList<Integer> liste2 = new ArrayList<Integer>();  
liste1 = liste2;
```
- ☒ Eine generische Klasse definiert potentiell viele Typen.
- ☐ Keine der Aussagen trifft zu
- ☒ Im Code sehen Sie die Typdeklaration `PatriciaTrie<Person>`. Sie schliessen daraus: `PatriciaTrie` ist eine generische Klasse.

Antwort speichern

➔ 1 Punkt

## Quizzies

Aktuelle Punktzahl: 2 / 13

Beantwortet: 2 / 14

Was verwenden Sie, wenn Sie in einer Klasse Rezeptbuch eine beliebige Anzahl von Objekten vom Typ Rezept speichern möchten?

- ☐ `ArrayList<String>`
- ☐ `String`
- ☒ `ArrayList<Rezept>`

Antwort speichern

➔ 1 Punkt

## Quizzies

Aktuelle Punktzahl: 3 / 13

Beantwortet: 3 / 14

Sie holen sich einen Iterator für eine wie folgt deklarierte Sammlung:

```
ArrayList<Auto> autos;
```

Welche Aussagen sind zutreffend?

- ☒ Im Gegensatz zu einer foreach Schleife, können beim Iterieren über die Sammlung mit einem Iterator während dem Iterieren auch Elemente entfernt werden.
- ☒ Löschen von Elementen: Ein Iterator erlaubt nebst dem Iterieren einzig das Löschen des aktuellen Elements.
- ☐ Den Iterator hole ich mir mittels: `Iterator<String> iter = autos.iterator();`
- ☐ Mit einem Iterator können Sie in beliebiger Reihenfolge über die in der ArrayList gespeicherten Elemente iterieren.
- ☒ Den Iterator hole ich mir mittels: `Iterator<Auto> iter = autos.iterator();`

Antwort speichern

➔ 1 Punkt

## Quizzies

Aktuelle Punktzahl: 4 / 13

Beantwortet: 4 / 14

Eine Klasse hat ein wie folgt deklariertes Datenfeld:

```
private ArrayList<Auto> autos;
```

Der Code im Konstruktor der Klasse lautet:

```
autos.add(new Auto()); //1  
autos.add(new Auto()); //2
```

Welche der folgenden Aussagen sind korrekt:

- ☐ Bei Ausführung von Zeile 2 tritt ein Fehler auf (IndexOutOfBoundsException)
- ☒ Bei Ausführung von Zeile 1 tritt ein Fehler auf (NullPointerException).
- ☒ Die ArrayList muss zuerst erzeugt werden, sonst hat das Datenfeld den Wert null.

Antwort speichern

➔ 0 Punkte

- Auch beim zweiten Versuch (keine Auswahl) ➔ 0 Punkte

## Quizzies

Aktuelle Punktzahl: 4 / 13

Beantwortet: 5 / 14

Eine foreach-Schleife iteriert über alle Elemente in einer Sammlung und sieht für eine Sammlung list mit Elementen vom Typ String wie folgt aus:

```
for(String text : list) {  
    // text zeigt auf das aktuelle Element  
}
```

☒ Trifft zu

☐ Trifft nicht zu

Antwort speichern

➔ 1 Punkt

## Quizzies

Aktuelle Punktzahl: 5 / 13

Beantwortet: 6 / 14

Mit den Codezeilen

```
autos.add(new Auto());  
System.out.println(new String("2+3=5"));
```

☒ werden zwei anonyme Objekte erzeugt.

☒ wird wahrscheinlich ein mittels des parameterlosen Konstruktors erzeugtes Auto-Objekt einer Sammlung hinzugefügt.

☒ wird genau ein anonymes Objekt erzeugt.

Antwort speichern

➔ 0 Punkte

○ Auch beim zweiten Versuch (Auswahl b) und c)) ➔ 0 Punkte

## Quizzies

Aktuelle Punktzahl: 5 / 13

Beantwortet: 7 / 14

Eine ArrayList ist eine Sammlung flexibler Grösse.

☒ Ja

☐ Nein

Antwort speichern

➔ 1 Punkt

## Quizzies

[Test beenden](#)[Test unterbrechen](#)

Aktuelle Punktzahl: 6 / 13

Beantwortet: 8 / 14

Die Klasse Ball modelliert einen Ball und verfügt über einen Konstruktor ohne Parameter. Der folgende Code erzeugt eine neue ArrayList zum Speichern von Ball Objekten und befüllt die Liste anschliessend mit zwei Ball Objekten. Ist das korrekt?

```
ArrayList<Ball> list;  
list = new ArrayList<Ball>();  
list.add(new Ball());  
list.add(new Ball());
```

☐ Nein

☒ Ja

Antwort speichern

[nach oben](#)

➔ 1 Punkt

## Quizzies

[Test beenden](#)[Test unterbrechen](#)

Aktuelle Punktzahl: 7 / 13

Beantwortet: 9 / 14

Sie haben eine ArrayList list vom Typ ArrayList<Integer> vor sich, aus der Sie alle Elemente entfernen wollen, deren Wert grösser ist als 100. Welche Schleifenköpfe eignen sich für diese Aufgabe?

☒ Iterator<Integer> it = list.iterator();  
while( it.hasNext() )

☐ for( Integer i : list )

Antwort speichern

[nach oben](#)

➔ 1 Punkt

Beantwortet: 10 / 14

Fügt man ein Auto-Objekt mit der Farbeigenschaft Blau in zwei unterschiedliche ArrayList ein und ändert anschliessend die Farbe aller Autos in der ersten ArrayList auf Rot, dann wechselt mindestens ein Auto in der zweiten ArrayList auf Rot.

☐ Nein

☒ Ja

Antwort speichern

[nach oben](#)

➔ 1 Punkt

## Quizzies

Aktuelle Punktzahl: 9 / 13

Beantwortet: 11 / 14

meineListe ist vom Typ ArrayList<String>. Die folgende Codezeile: meineListe.get(meineListe.size())

☒ führt zu einer IndexOutOfBoundsException.

☒ gibt das letzte Element in der Liste zurück oder führt zu einer IndexOutOfBoundsException falls die ArrayList keine Elemente enthält.

Antwort speichern

➔ 0 Punkte

- Zweiter Versuch korrekt ➔ korrekte Antwort «IndexOutOfBoundsException»

## Quizzies

[Test beenden](#)[Test unterbrechen](#)

Aktuelle Punktzahl: 9 / 13

Beantwortet: 12 / 14

Eine ArrayList ist generisch: Sie kann im Prinzip für beliebige Datentypen erzeugt werden. Also kann eine ArrayList auch selbst wiederum ArrayList Objekte enthalten. Diese würde man dann z.B. wie folgt erzeugen:

```
ArrayList<ArrayList<String>> list = new ArrayList<ArrayList<String>>();
```

☒ Nein, eine Liste kann keine andere Liste enthalten

☐ Ja

☐ Nein, >> ist ein Operator und es geht deshalb nicht.

[Antwort speichern](#)

➔ 0 Punkte

- Zweiter Versuch korrekt ➔ korrekte Antwort «Ja»