



Schiffe versenken – Wiederholung Grundlagen

22. November 2024

Plan für die Woche

Montag

- Syntax und Grundelemente
- Kontrollstrukturen
- Klassen und Objekte

Dienstag

- Git Workflow
- Umgang mit großen Aufgabenstellungen
- Arrays

Mittwoch

- Schiffe versenken

Donnerstag

- Methoden
- OOP
- Einführung in UML

Freitag

- ArrayList

Projekt: Schiffe versenken

Eigene Schiffe

	1	2	3	4	5	6	7	8	9	10	
A						4		3			A
B	1					4					B
C						4					C
D			3								D
E									2		E
F					2		3				F
G											G
H	4		4								H
I	4									1	I
J						3					J
	1	2	3	4	5	6	7	8	9	10	

Gegnerische Schiffe

	1	2	3	4	5	6	7	8	9	10	
A											A
B											B
C											C
D											D
E											E
F											F
G											G
H											H
I											I
J											J
	1	2	3	4	5	6	7	8	9	10	



Plan für heute

- Üben Klassendiagramme zu erstellen
- Wiederholung JavaDoc
- ArrayList



UML-Klassendiagramm

Erstelle ein passendes Klassendiagramm für die Modellierung eine Bibliothek.

- Es muss eine Klasse *Bibliothek* geben.
- Eine Bibliothek enthält verschiedene, daher muss es auch eine Klasse *Medien* geben.
 - Sie soll den Titel, die ISBN, das Erscheinungsjahr und den Status (verfügbar/nicht verfügbar) enthalten.
 - Und die Methoden ausleihen() und zurueckgeben()
- Überlege dir mindestens drei Subklassen, die von Medien erben könnten. Überlege dir, welche Attribute noch fehlen und ob noch weitere Methoden fehlen.



Wiederholung JavaDoc

ArrayList

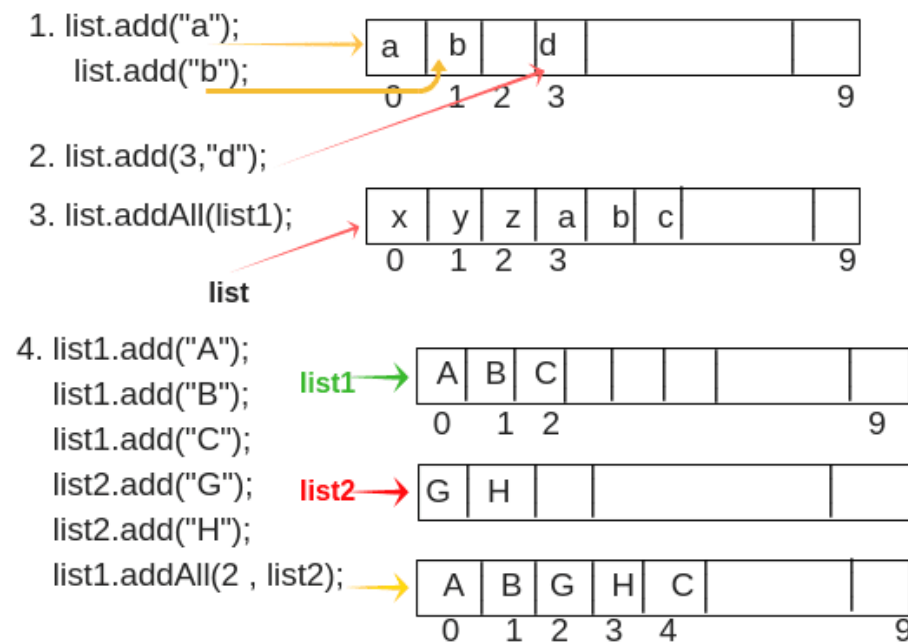
Collections

- Java Framework
- Umfasst Gruppe von Datenstrukturen, zum speichern, verwalten und manipulieren von Daten
- Stellt verschiedene Interfaces (Schnittstellen) und Klassen zur Verfügung: Listen, Maps und Sets

Details siehe auch: <https://docs.oracle.com/javase/8/docs/api/?java/util/Collections.html>

Collections: List

- Geordnete Sammlung, in der die Elemente nach ihrer Einfüge-Reihenfolge aufgerufen werden
- ArrayList, LinkedList...





Collections: List

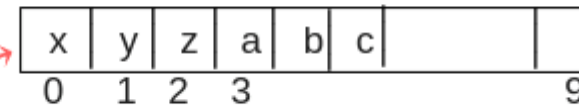
Wie muss der Befehl lauten, um an Position 6 den char „d“ hinzuzufügen?

- a) `list.add(5, „d“)`
- b) `list.add(6, „d“)`
- c) `list.add(d, 6)`
- d) `list.add(„d“, 6)`

2. `list.add(3, "d");`

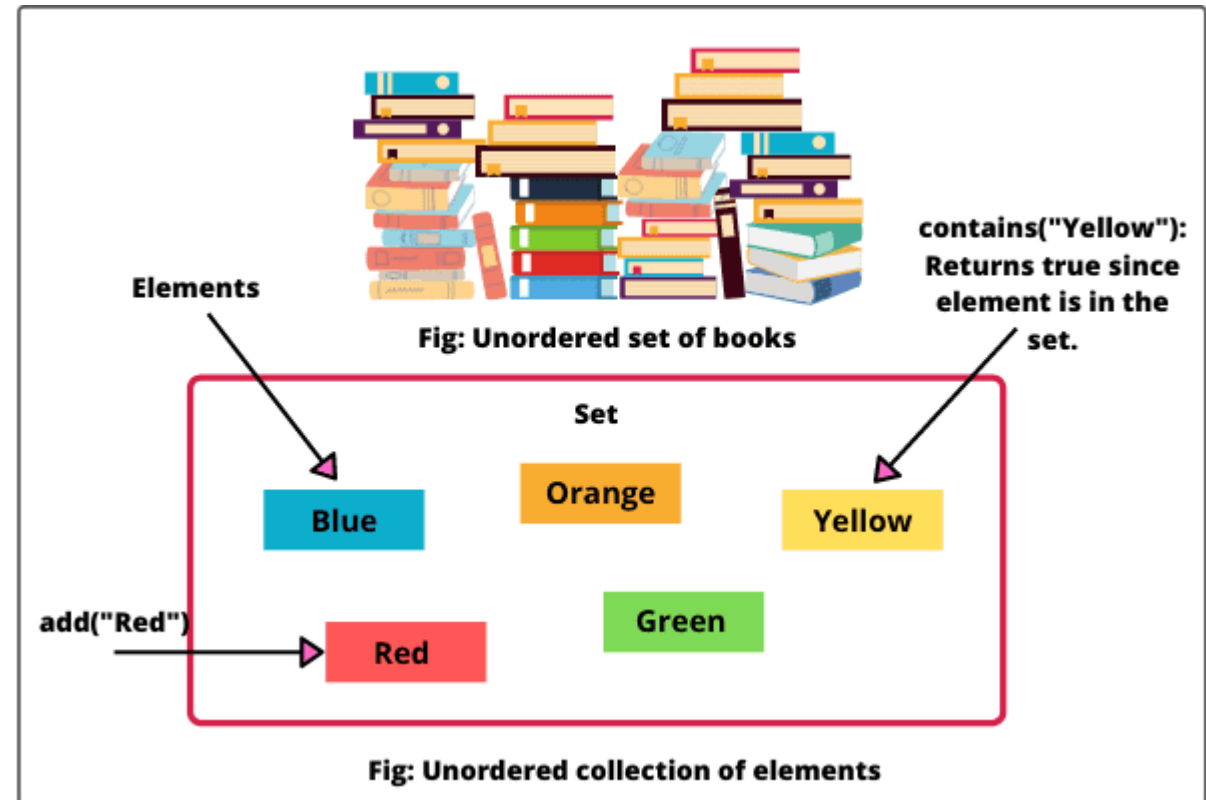
3. `list.addAll(list1);`

`list`



Collections: Set

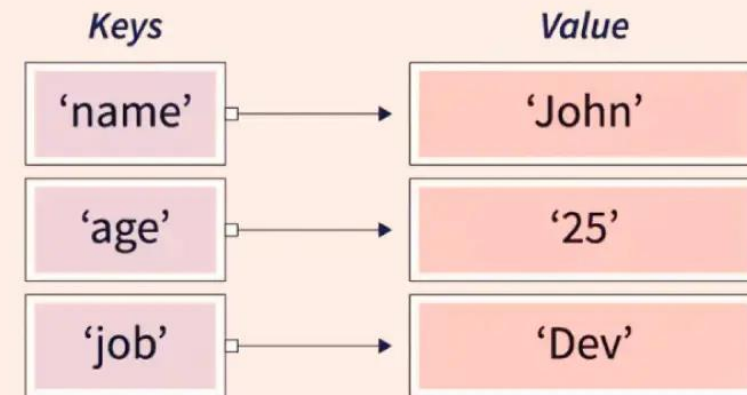
- Ungeordnete Sammlung
- Lässt keine Duplikate zu
- HashSet, TreeSet



Collections: Map

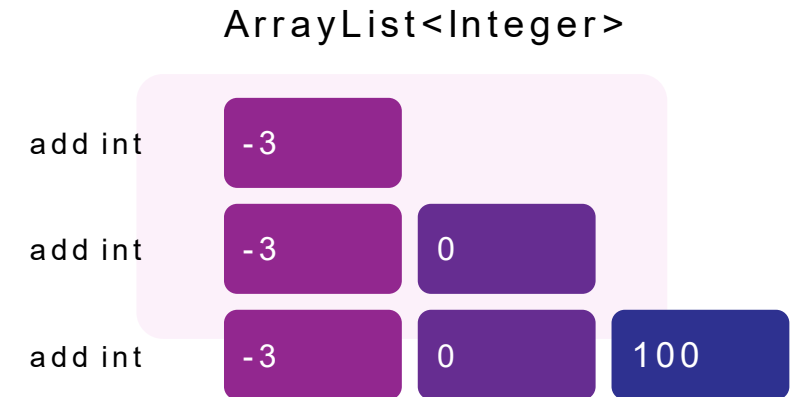
- Sammlung von Schlüssel-Wert-Paaren
- HashMap, TreeMap

What is Java Map?



ArrayList

- Sammlung von Objekten
- Java-Standardbibliothek und Teil des java.util-Paket
- Automatische Anpassung ihrer Größe
- Jedes Element hat eine Position (Index)



ArrayList

Verschiedene Möglichkeiten der Deklaration:

```
// Eine ArrayList von Strings erstellen
ArrayList<String> list = new ArrayList<>(); 1 usage

//Beispiele: Verschiedene Varianten der Deklaration, die auch in den Test zu finden sein werden
//TODO Korrekter Import
List list = new ArrayList(); 1 usage

//Copy an existing list
ArrayList<String> stringList = new ArrayList<String>(list); no usages

//List is of type Students
List<Student> list1 = new ArrayList<>(); no usages
```


ArrayList

Was kann man auf einer ArrayList machen?

- hinzufügen: *add(E Element)*
- abrufen: *get(int index)*
- Größe bestimmen: *size()*
- entfernen: *remove(Objekt o)* oder *remove(int index)*
- enthält: *contains(Objekt o)*
- alle Elemente entfernen: *clear()*



Aufgabe

1. Erstelle eine Obstkiste als ArrayList. Es sollen mindestens 3 Obstsorten hinzugefügt werden. Gebe den Stand in der Konsole aus.
2. Gib dann das erste Element der Liste aus.
3. Entferne eine Frucht und gebe die neue Obstkiste aus.
4. Bestimme die Größe der Obstkiste und gebe diese aus.
5. Überprüfe, ob die Kiste ein bestimmtes Obst enthält.
6. Leere die Liste und gebe den Stand aus.

10 min

ArrayList vs Array

Kriterium	Array	ArrayList
Größe	Feste Größe	Dynamische Größe
Speicherbedarf	Geringer Speicherbedarf	Höherer Speicherbedarf
Zugriffszeit	Sehr schnell	Schnell
Hinzufügen und Löschen von Elementen	Nicht flexibel	Flexibel
Verwendbarkeit	Statisch	Dynamisch
Leistung bei Änderungen	Langsam	Schneller bei Änderungen
Flexibilität bei Datentypen	Wenig flexibel	Hohe Flexibilität
Eingebaute Methoden	Keine Methoden	Viele Methoden

Quellen

<https://www.shiksha.com/online-courses/articles/implementing-array-in-java/>

<https://codegym.cc/de/groups/posts/789-matrix-in-java-2d-arrays>

<https://www.scientecheasy.com/2020/09/java-list-interface.html/>

<https://www.scientecheasy.com/2020/10/java-set.html/>

<https://www.designgurus.io/blog/what-is-java-map-class-and-its-uses>

<https://www.dotnetperls.com/arraylist-integer-java>