

Selbststudium OOP5

Auftrag

Kapitel 4.1 bis 4.3

1. zu bearbeitende Aufgaben: 4.1 – 4.3

4.1:

done

4.2:

Es gibt keinen Error. Die Methode removeFile() überprüft ob ein gültiger Index übergeben wurde.

4.3:

Es wird der Dateiname der zweiten eingefügten Datei zurückgegeben.

In einer Collection werden nachfolgenden Objekte nach vorne verschoben wenn ein Element davor gelöscht wird.

2. Was verstehen Sie unter einem "Java-Package"?

Ein Java-Package ist eine Ansammlung von Klassen, welche in Bibliotheken zur Verfügung stehen.

3. Sie möchten in Ihrem Source-Code die Bibliotheksklasse ArrayList verwenden. Mit welcher Anweisung können Sie die Bibliotheksklasse ArrayList verfügbar machen?

```
import java.util.ArrayList;
```

Kapitel 4.4 bis 4.7

4. zu bearbeitende Aufgaben: 4.4 bis 4.7

4.4:

```
ArrayList<Book> library;  
library = new ArrayList<Book>();
```

4.5:

```
ArrayList<Student> cs101;
```

4.6:

```
private ArrayList<MusicTrack> tracks;
```

4.7:

```
private ArrayList<Book> library = new ArrayList<>();  
private ArrayList<Book> library = new ArrayList<Book>();  
ArrayList<Student> cs101 = new ArrayList<>();  
ArrayList<Student> cs101 = new ArrayList<Student>();  
private ArrayList<MusicTrack> tracks = new ArrayList<>();  
private ArrayList<MusicTrack> tracks = new ArrayList<MusicTrack>();
```

5. zu bearbeitende Aufgaben: 4.8 bis 4.11

4.8:

size gibt die Anzahl Objekte zurück: 10.

4.9:

```
element.get(4); // holt das 5 Element, da der Index bei 0 beginnt
```

4.10:

14 (Anzahl Elemente – 1)

4.11:

```
files.add(favoriteTrack);
```

6. zu bearbeitende Aufgaben: 4.12 und 4.13

4.12:

```
dates.remove(2); // löscht das 3. Element
```

4.13:

Der neue Index ist 5.

7. Erläutern Sie die nachfolgende Deklaration: `private ArrayList<Balloon> list = new ArrayList<>();`

Es ist eine private ArrayList für Objekt-Typ Balloon, der Variablenname ist list.

8. Inwiefern haben Abstraktion und ArrayList miteinander zu tun?

Unter Abstraktion versteht man das „verstecken, vereinfachen“ von Code, resp. deren Verwendung.

Beim ArrayList wird sehr viel Code „versteckt“, resp. die Verwendung der ArrayList durch die Methoden vereinfacht.

9. Inwiefern unterscheiden sich die Methoden `remove()` und `get()` von ArrayList?

remove() entfernt das Item aus der Liste, get() liefert den Inhalt des Items zurück.

Kapitel 4.8. bis 4.12

10. zu bearbeitende Aufgaben: 4.18 und 4.19

4.18:

```
public void listAllFiles() // falls die Filenamen direkt ausgegeben werden  
public ArrayList listAllFiles() // falls die Filenamen zurückgegeben werden
```

Beide Möglichkeiten benötigen keine Parameter.

4.19:

Genauso viele wie es Einträge in der ArrayList gibt.

11. zu bearbeitende Aufgaben: 4.22

done

12. Erläutern Sie möglichst ausführlich den Source-Code auf Seite 108.

Dieser Code iteriert mit der foreach-Schleife über alle Elemente des Typs String der ArrayList files.

Diese Elemente bzw. Strings werden dann jeweils auf die Konsole ausgegeben. (für jeden filename in files gib den Inhalt von filename aus)

13. Kann es passieren, dass der Schleifen-Rumpf einer while-Schleife gar nie zur Ausführung gelangt?

Ja, falls die Bedingung zu Beginn schon nicht true ist.

14. Geben Sie für `no++` zwei alternative Anweisungen an.

```
no = no + 1;  
no += 1;
```

15. Eine ArrayList lässt sich mit Hilfe einer foreach-Schleife traversieren. Kennen Sie noch weitere Möglichkeiten?

Mit dem Iterator, oder mittels for / while / do-while Schleifen.

16. Ist `hasNext()` eine Methode von ArrayList oder Iterator? Wie ist der Rückgabewert von `hasNext()` zu interpretieren?

hasNext() ist eine Methode vom Iterator.

hasNext() liefert ein Boolean zurück ob es ein weiteres Objekt in der Liste gibt welches noch nicht abgerufen wurde.

Kapitel 4.14

17. Sie können diese Kapitel getrost überspringen. Lesen Sie nur die ‚Concept‘ Box auf Seite 130.

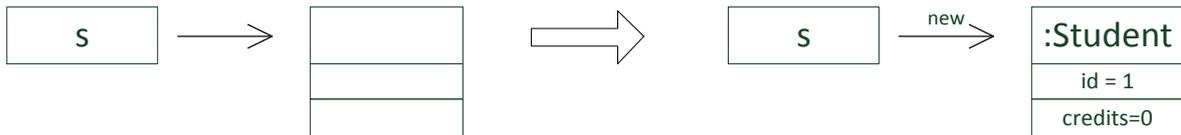
done

18. Eine Variable, deklariert für einen Klassentyp (Eine solche Variable wird auch Referenzvariable genannt), kann den Spezialwert null beinhalten. Veranschaulichen diesen Fall mit einer Zeichnung. Wie sieht es aus, wenn die Variable ein Objekt speichert?

Bei der Deklaration wird erst der Speicherbereich für das Objekt reserviert. Die Referenzadresse zeigt erst auf den Speicherplatz wo später das Objekt gespeichert wird, zu diesem Zeitpunkt besitzt sie noch den Wert null.

Erst wenn das Objekt instanziiert wird, wird im Speicher das Objekt erzeugt. In der Variable steht noch immer nur die Referenzadresse auf das Objekt.

```
private Student s;
s = new Student();
```



Kapitel 4.15 bis 4.17

19. zu bearbeitende Aufgaben: 4.62 bis 4.65

4.62:

```
private Person[] people;
```

4.63:

```
private boolean[] vacant;
```

4.64:

2x hourCounts

Variable, Array Init

1x hourCounts.length

Anzahl Array Elemente abfragen

2x hourCounts[hour]

Counter der selektierten Stunde (hour) erhöhen, resp. abfragen

4.65:

Falsch:

```
[]int counts;
```

Richtig:

```
int[] counts;
```

```
boolean[5000] occupied;
```

```
boolean[] occupied;
```

```
occupied = new boolean[5000];
```

oder:

```
boolean[] occupied = new boolean[5000];
```

20. zu bearbeitende Aufgaben: 4.66 bis 4.68

4.66:

```
double[] readings = new double[6];
String[] urls = new String[9];
TicketMachine[] machines = new TicketMachine[5];
```

4.67:

20 String Objekte

4.68:

```
double[] prices = new double[50];
```

21. Was spricht für und was spricht gegen Arrays?

Positiv:

- *Dynamische Grösse*
- *Performance*

Negativ:

- *Statische Grösse*

22. Wie lässt sich die Länge eines Arrays ermitteln?

```
array.length();
```

23. zu bearbeitende Aufgaben: 4.69, 4.71, 4.73 und 4.74

4.69:

ArrayIndexOutOfBoundsException

4.71:

```
public void printGreater(double marks[], double mean)
{
    for(int index = 0; index < marks.length; index++) {
        if(marks[index] > mean) {
            System.out.println(marks[index]);
        }
    }
}
```

4.73:

```
public int numberOfAccesses()
{
    int total = 0;
    int[] hourCounts = new int[24]; // pro Stunde
    for(int i = 0; i < hourCounts.length; i++) {
        total += hourCounts[i];
    }
    return total;
}
```