

## PR1 – Formular für Lesenotizen

### WS2020/21

Nachname Abdel Kader	Vorname Schehat	Matrikelnummer 1630110	Abgabedatum: 03.12.2020
-------------------------	--------------------	---------------------------	----------------------------

### Arrays (L.7) & Collections (L.8.1-8.2)

#### Lernzielfragen:

- a) Schreiben Sie eine Methode, die jedes dritte Element eines als Parameter gegebenen double-Arrays auf den Wert seines Vorgängerelementes setzt

```
import java.util.ArrayList;
public class La {
    public static void main(String[] args) {
        ArrayList<Double> numbers = new ArrayList<Double>();
        for (double i = 0; i < 10; i++) {
            numbers.add(i);
        }
        for (int i = 0; i < numbers.size(); i++) {
            if ((i+1) % 3 == 0) {
                numbers.set(i, numbers.get(i-1));
            }
        }
    }
}
```

- b) Verwenden Sie die Methode split der Klasse String, um ein Datum aus dem amerikanischen Format (Monat/Tag/Jahr) in das deutsche Format (Tag.Monat.Jahr) zu überführen

```
import java.util.ArrayList;
public class Lb {
    public static void main(String[] args) {
        String s = "Monat/Tag/Jahr";
        String[] arrS = s.split("/");
        s = arrS[0];
        for (int i = 1; i < arrS.length; i++) {
            s = s + "." + arrS[i];
        }
    }
}
```

- c) Woran scheitert das folgende Programmfragment? Begründen Sie?

```
String[] w= new String[4];
for (int i=1; i<4; i++) {
    w[i]= String.valueOf(w[i-1].length());
}
```

da der Array w null Objekte als Elemente besitzt und versuch auf die Methode length zuzugreifen, jedoch besitzt null diese Methode nicht: NullPointerException

- d) Wie können Sie mit den Methoden der Klasse ArrayList herausfinden, ob ein Wert mehrfach in einer gegebenen Liste vorkommt? Und zwar ohne eine eigene Schleife zu programmieren

```
if (numbers.indexOf(<value>) != numbers.lastIndexOf(<value>))
```

**Array**

Deklaration: `<type>[] <name> = new <type>[<length>]`

Automatische Initialisierung: `int: 0, double: 0.0, boolean: false, char: '\0', Objekte: null`

Schreibender Zugriff: `<name>[<index>] = <value>`

Lesender Zugriff: `<name>[<index>]`

Index: gültige Länge von 0 bis Arraylänge - 1

Array Länge: `<name>.length` ! keine Klammern

Initialisierung: `<type>[] <name> = {<value1>, ...}`

Array als Rückgabewert: `public static <type>[] <name> (...) & Array returnen`

Array als Eingabeparameter: `public static <type> <name> (<type>[] <Array name>)`

Array als Ausgabeparameter: Array sind Objekte => Zeigersemantik, also kein Rückgabewert nötig um einen Array zu ändern

**Klasse Array:** `import java.util.Array`

Methode	Bedeutung
<code>binarySearch(array, wert)</code>	Index von wert zurück (oder -1) & Array muss sortiert sein
<code>equals(array1, array2)</code>	true, wenn Arrays gleiche Elemente in gleicher Reihenfolge
<code>fill(array, wert)</code>	Setzt jedes Element im Array auf den gegebenen Wert
<code>sort(array)</code>	Sortiert Elemente von Array in aufsteigender Reihenfolge
<code>toString(array)</code>	Liefert eine Zeichenkette für die Ausgabe
<code>compare(array1, array2)</code>	Lexikografischer Vergleich

**Kommandozeilenargumente:** args Array übergebener Parameter**NullPointerException:** keine Methoden von null Objekten aufrufen**Zweiphasen-Initialisierung von Objekten:** 1. Array erzeugen & 2. new <type Objekt> hinzufügen

Weitere String Methoden	Beschreibung	Beispiel
<code>toCharArray()</code>	Arrays von chars	String s = "long book"; ⇒ {'l', 'o', 'n', 'g', ' ', 'b', 'o', 'o', 'k'}
<code>split(Begrenzer)</code>	Separiert String zwischen Begrenzer	s.split(" ") liefert {"long", "book"}
<code>String.join(Begrenzer,array)</code>	Setzt Elemente im Array als String	arr = {"a", "b", "c"} : String.join("-", arr) liefert "a-b-c"

**Klasse ArrayList<E>:** `import java.util.ArrayList`

- Eigenschaften: Anzahl Elemente beliebig, bietet Methoden & <E> steht für Datentyp (z.B.: Integer)

Deklaration: `ArrayList<type> <name> = new ArrayList<type>()`

Methode
<code>add (value)</code>
<code>add (index, value)</code>
<code>clear ()</code>
<code>contains (value)</code>
<code>get (index)</code>
<code>indexOf (value)</code>
<code>lastIndexOf (value)</code>
<code>remove (index)</code>
<code>set (index, value)</code>
<code>size ()</code>

Collection: Behälter zur Speicherung von Daten

ArrayList: Größenveränderbarer Array
HashSet: Menge (keine Duplikate)
TreeSet: sortierte Menge (keine Duplikate)
HashMap: Abbildung (Werte-Paare)
TreeMap: sortierte Abbildung

Methode
<code>add (value)</code>
<code>addAll (collection)</code>
<code>remove (value)</code>
<code>clear ()</code>
<code>contains (value)</code>
<code>containsAll (collection)</code>
<code>isEmpty ()</code>
<code>removeAll (collection)</code>
<code>retainAll (collection)</code>
<code>size ()</code>
<code>toArray ()</code>
<code>iterator ()</code>