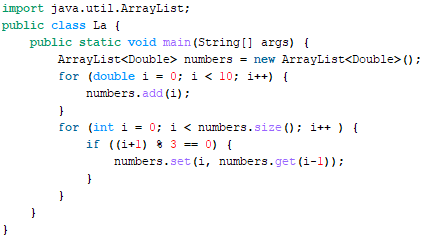
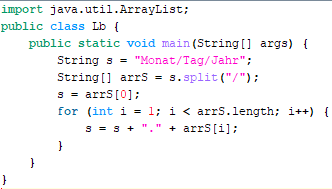
**PR1 – Formular für Lesenotizen**

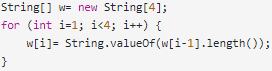
**WS2020/21**

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

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

**Lernzielfragen:**

1. Schreiben Sie eine Methode, die jedes dritte Element eines als Parameter gegebenen double-Arrays auf den Wert seines Vorgängerelementes setzt
2. 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
3. Woran scheitert das folgende Programmfragment? Begründen Sie?

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

1. 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

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

**Kommandozeilenargumente:** args Array übergebener Parameter

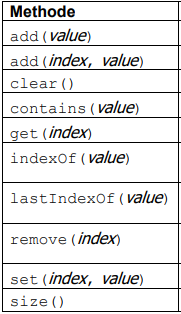
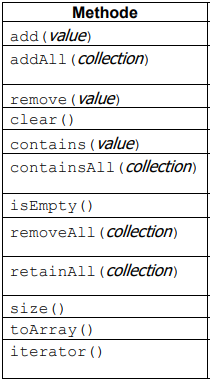
**NullPointerException:** keine Methoden von null Objekten aufrufen

**Zweiphasen-Initialisierung von Objekten:** 1. Array erzeugen & 2. new <type Objekt> hinzufügen

|  |  |  |
| --- | --- | --- |
| **Weitere String Methoden** | Beschreibung | Beispiel  String s = "long book"; |
| toCharArray() | Arrays von chars | * {‘l‘, ‘o‘, ‘n‘, ‘g‘, …} |
| split(Begrenzer) | Separiert String zwischen Begrenzer | s.split(" ") liefert {"long", "book"} |
| String.join(Begrenzer,array) | 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>()

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 |