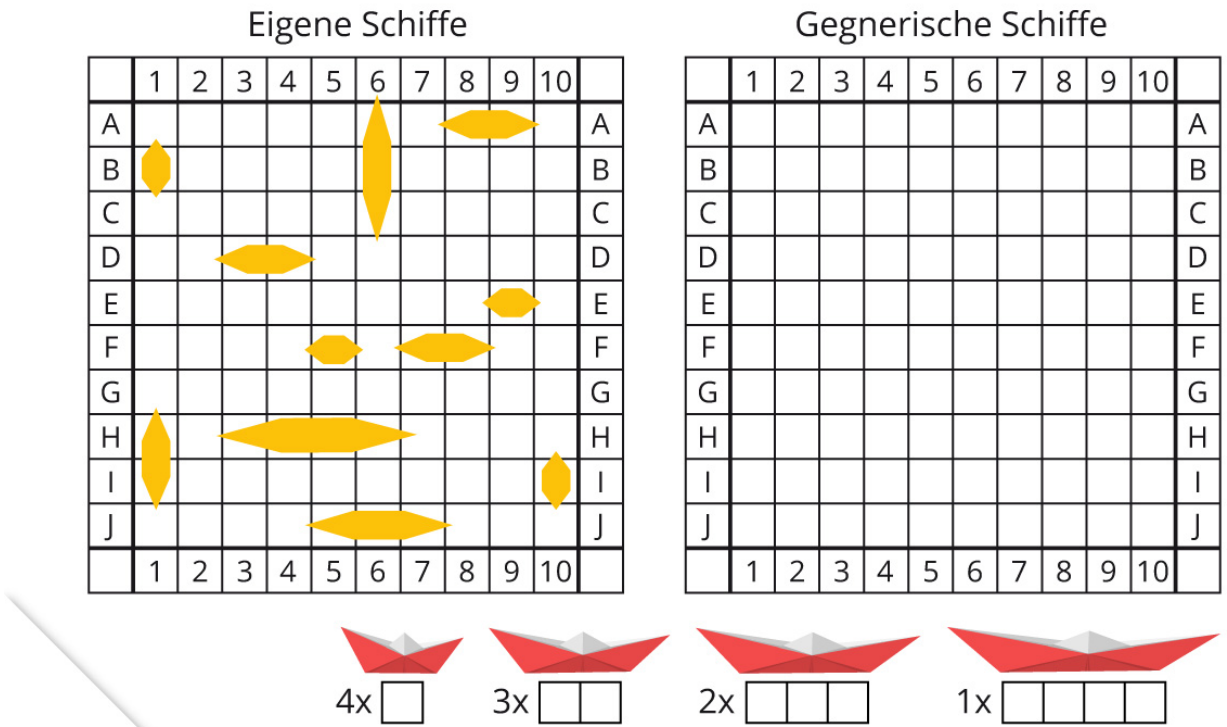
Schiffe versenken – Wiederholung Grundlagen

19. November 2024

Plan für die Woche

Montag	Dienstag	Mittwoch	Donnerstag	Freitag
• Syntax und Grundelemente • Kontrollstrukturen • Klassen und Objekte	• Git-Workflow • Umgang mit großen Aufgabenstellungen • Arrays • ArrayList	• Einführung in UML • Erste Grundlagen OOP	• Konsolen Ein- und Ausgabe	• Strategien Fehlererkennung und Debugging

Projekt: Schiffe versenken



https://www.talu.de/schiffe-versenken/

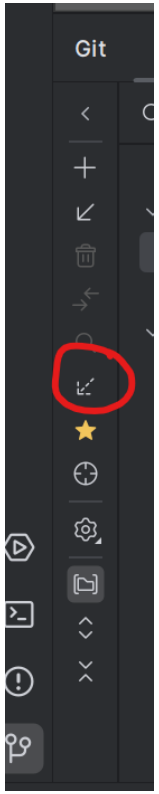
Plan für heute

- Git Workflow
- Umgang mit großen Aufgabenstellungen
- Arrays
- ArrayList

Git Workflow

Git Workflow

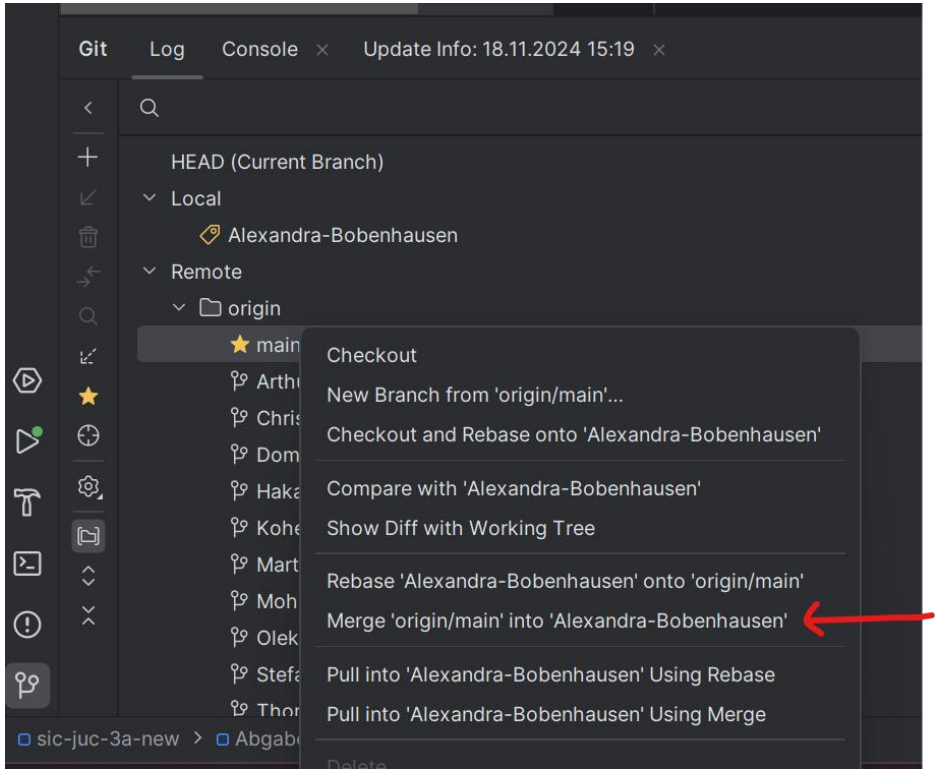
1. Intellij öffnen
2. Fetch



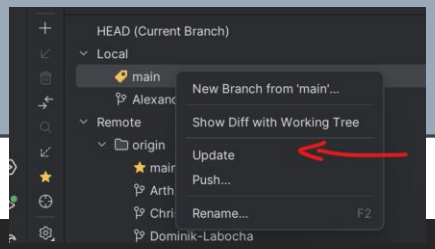
Git Workflow

3. Stelle sicher, dass du deinen Branch ausgecheckt hast!

4. Merge den origin/main-Branch in deinen Branch.



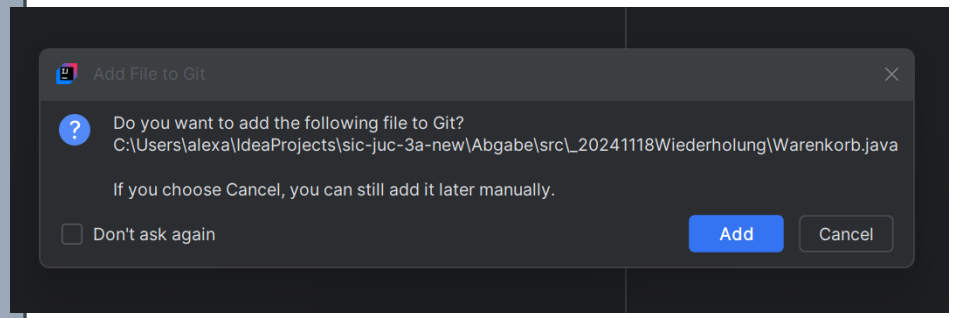
Wichtig: Wenn ihr lokal den main-Branch liegen habt, muss der vorher per „Update“ aktualisiert werden.



Git Workflow

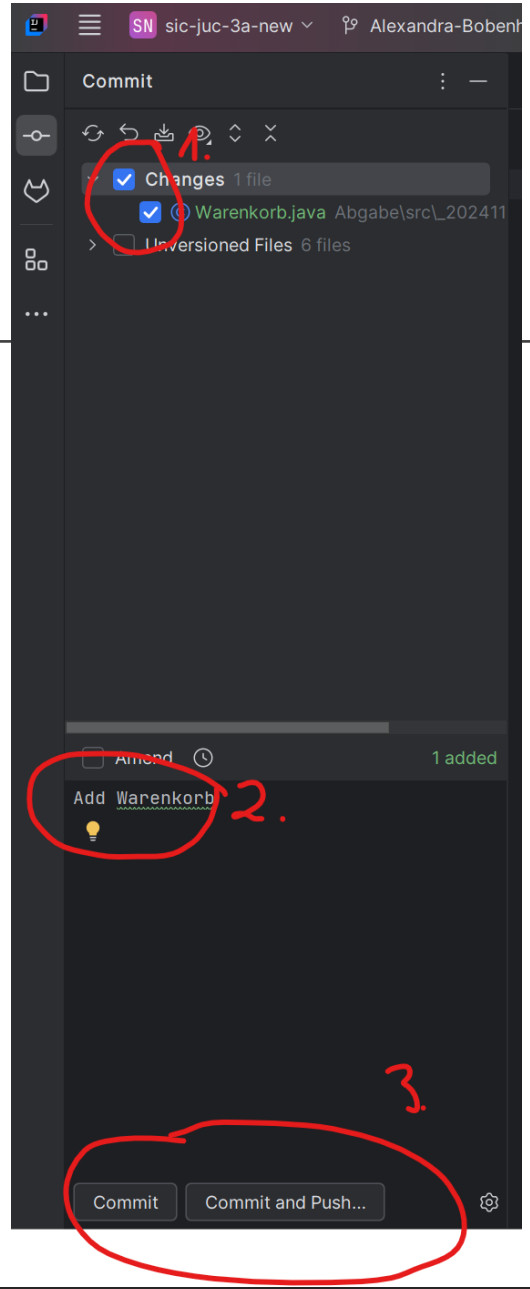
5. Beginne deine Arbeit.

Wichtig: Wenn du gefragt wirst, ob du ein File o.Ä. zu Git hinzufügen möchtest, musst du in der Regel immer mit „add“ bestätigen!



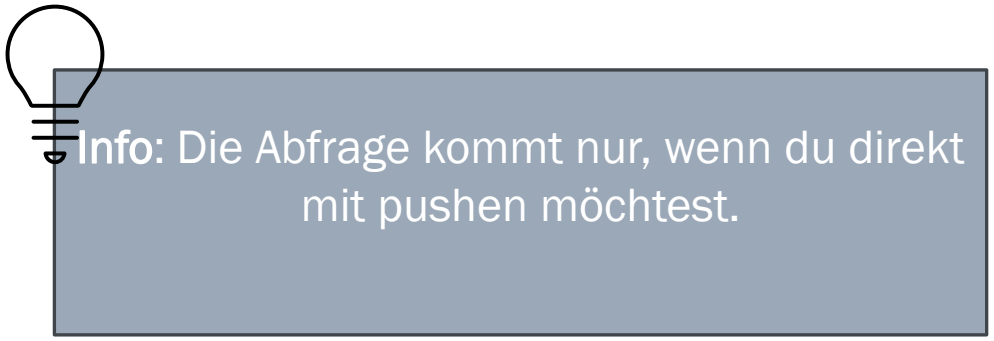
Git Workflow

6. Commite deine Zwischenstände.
   - Wähle deine Changes aus, die du commiten möchtest.
   - Schreibe eine passende und kurze Commit-Message.
   - **Commit**: Speichere im lokalen Repository. (Hier muss später noch in das Remote-Repository gepusht werden!)
   - **Commit and Push**: Du speicherst im lokalen und im remote Repository.

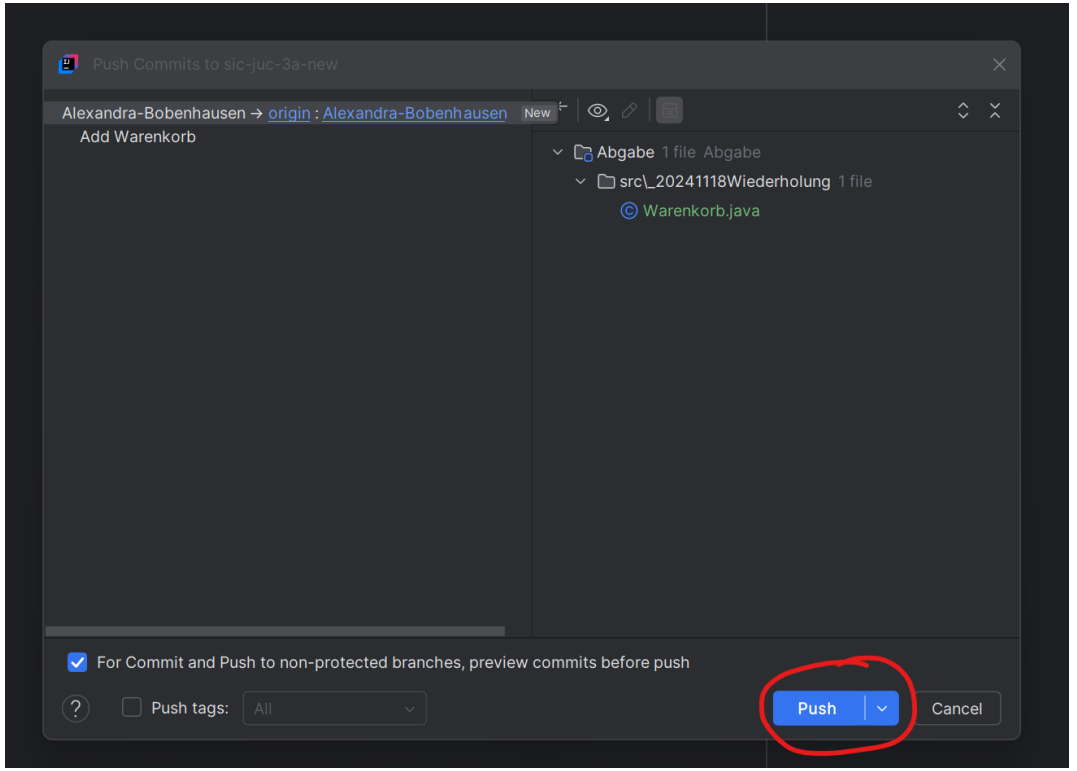


Git Workflow

7. Bestätige, was du commiten und pushen möchtest.



Info: Die Abfrage kommt nur, wenn du direkt mit pushen möchtest.



Umgang mit großen Aufgabenstellungen

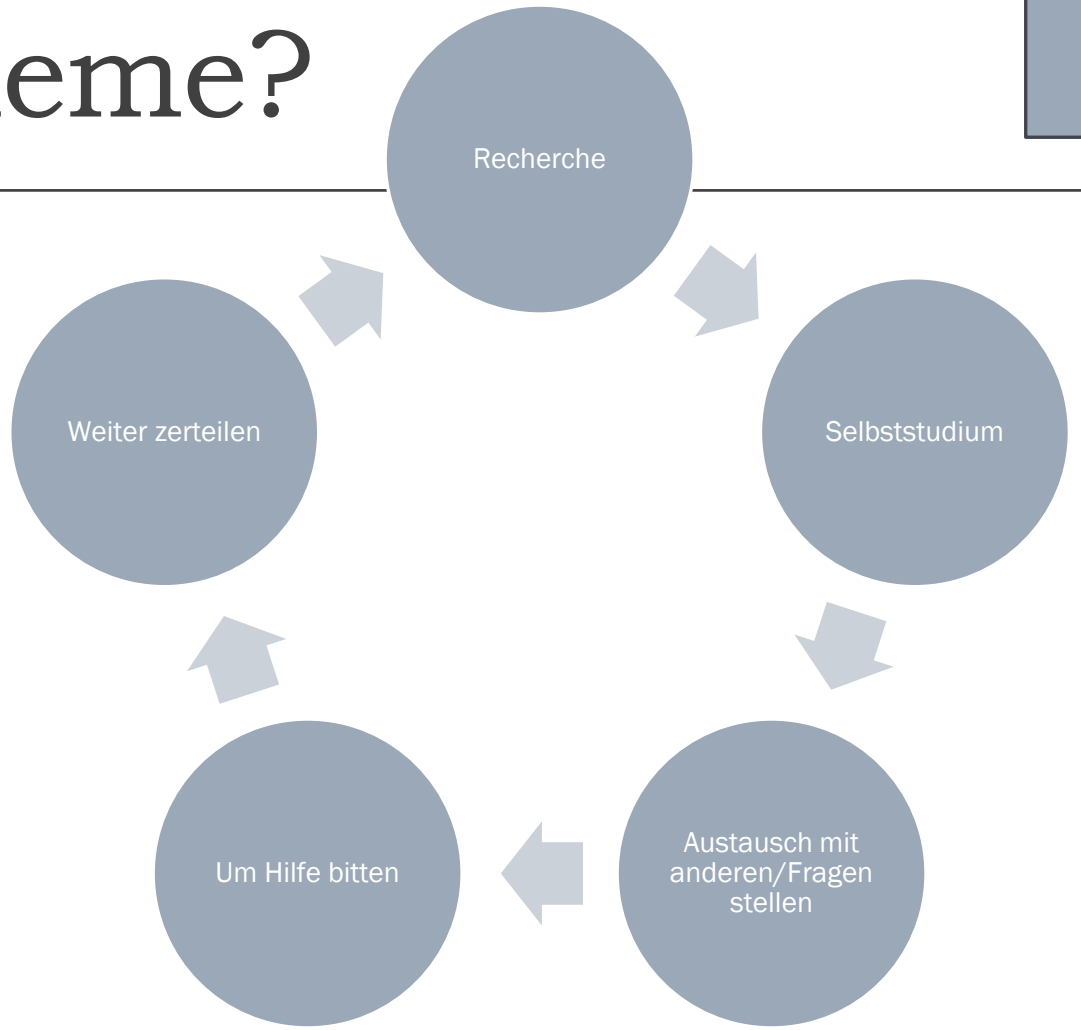
Diskussion

Wie gehst du vor, wenn du eine Aufgabe gestellt bekommst, um diese zu lösen?

Vorgehen


Lesen der Aufgabe
Markieren der Schlüsselwörter
Zerlegung in kleine Teilprobleme
Planung und Strukturierung
Umsetzung

Probleme?



Gute Seiten für Recherche und Selbststudium: Stack Overflow, GeeksforGeeks, W3Schools

Vorgehen erstellen

Erstelle eine Anwendung zur Verwaltung von Kontakten. Jeder Kontakt besteht aus einem **Namen**, einer **Telefonnummer** und einer **E-Mail-Adresse**. Du sollst eine **ArrayList** von Kontakten verwenden, um diese zu speichern und mit ihnen zu interagieren. Die Anwendung soll die folgenden Funktionen bieten:

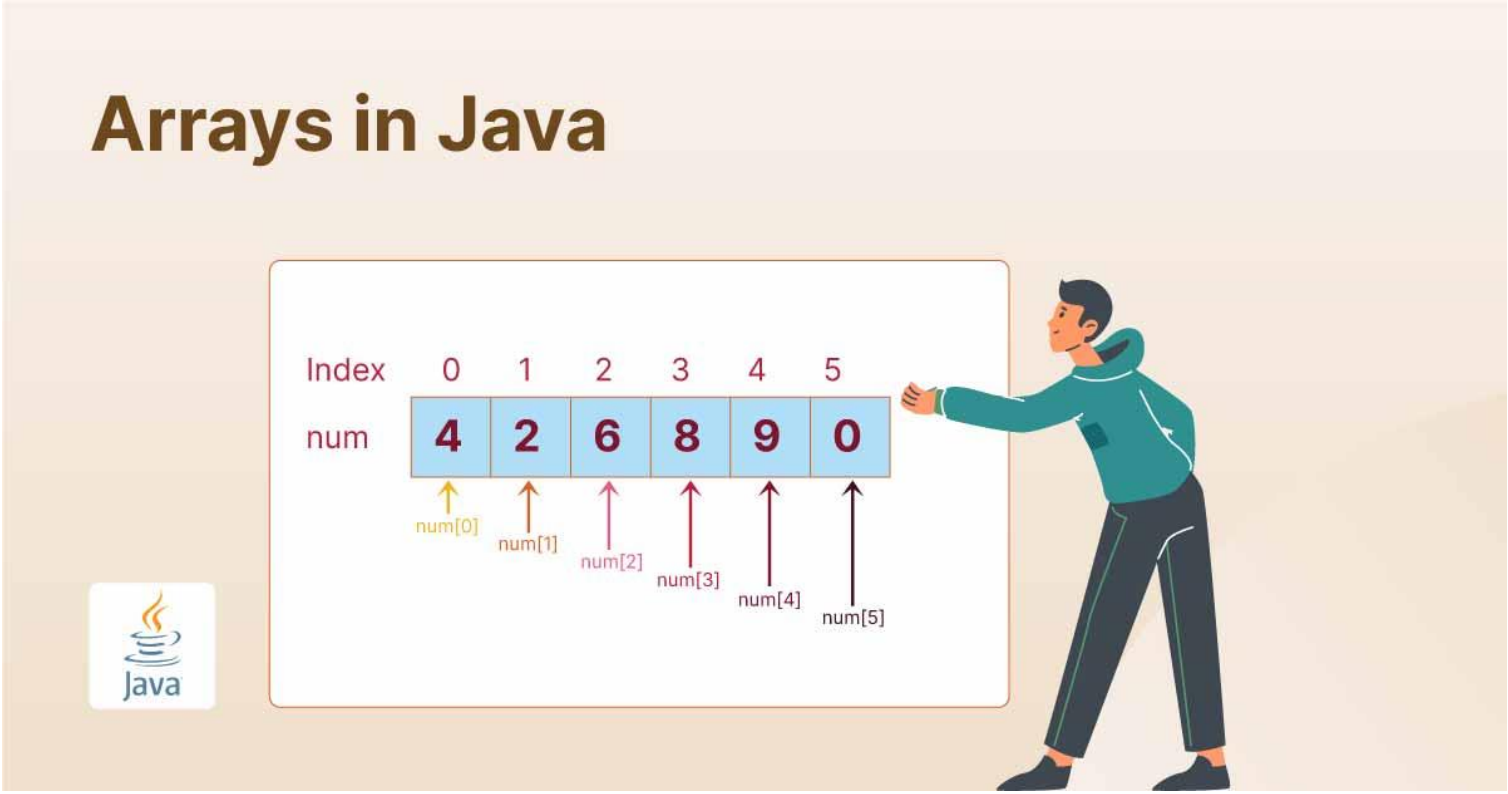
1. **Kontakt hinzufügen**:
   - Der Benutzer kann einen neuen Kontakt mit einem Namen, einer Telefonnummer und einer E-Mail-Adresse hinzufügen.
2. **Kontakte anzeigen**:
   - Der Benutzer kann alle gespeicherten Kontakte anzeigen, wobei jeder Kontakt mit seinem Namen, der Telefonnummer und der E-Mail-Adresse angezeigt wird.
3. **Kontakt suchen**:
   - Der Benutzer kann nach einem Kontakt anhand des Namens suchen. Die Anwendung soll den gefundenen Kontakt anzeigen oder eine Nachricht anzeigen, wenn der Kontakt nicht existiert.
4. **Kontakt bearbeiten**:
   - Der Benutzer kann die Telefonnummer oder E-Mail-Adresse eines bestehenden Kontakts ändern.
5. **Kontakt löschen**:
   - Der Benutzer kann einen Kontakt nach dem Namen löschen.
6. **Daten persistieren**:
   - Die Liste der Kontakte soll nach dem Beenden der Anwendung in einer Datei gespeichert und beim nächsten Start geladen werden.

Arrays

Arrays

- Sammlung von Elementen des **gleichen** Datentyps
- Feste Länge

Arrays


Arrays in Java
Index 0 1 2 3 4 5
num 4 2 6 8 9 0
num[0]
num[1]
num[2]
num[3]
num[4]
num[5]
Java

Arrays

- **Deklaration:** Datentyp [] nameDesArrays
- **Initialisierung:**
  - Direkt befüllen: `int[] arrayName = {1, 2, 3, 4, 5};`
  - Mit new-Operator einen leeren Array erstellen: `int[] arrayName = new int[5];`

Arrays

- Zugriff auf Element eines Arrays über Indizes – beginnend bei 0

```
arrayName[0]
```

- Länge eines Arrays

```
arrayName.length
```

Arrays

Wichtigste Operationen auf einem Array:

- arrayName.length
- arrayName.clone

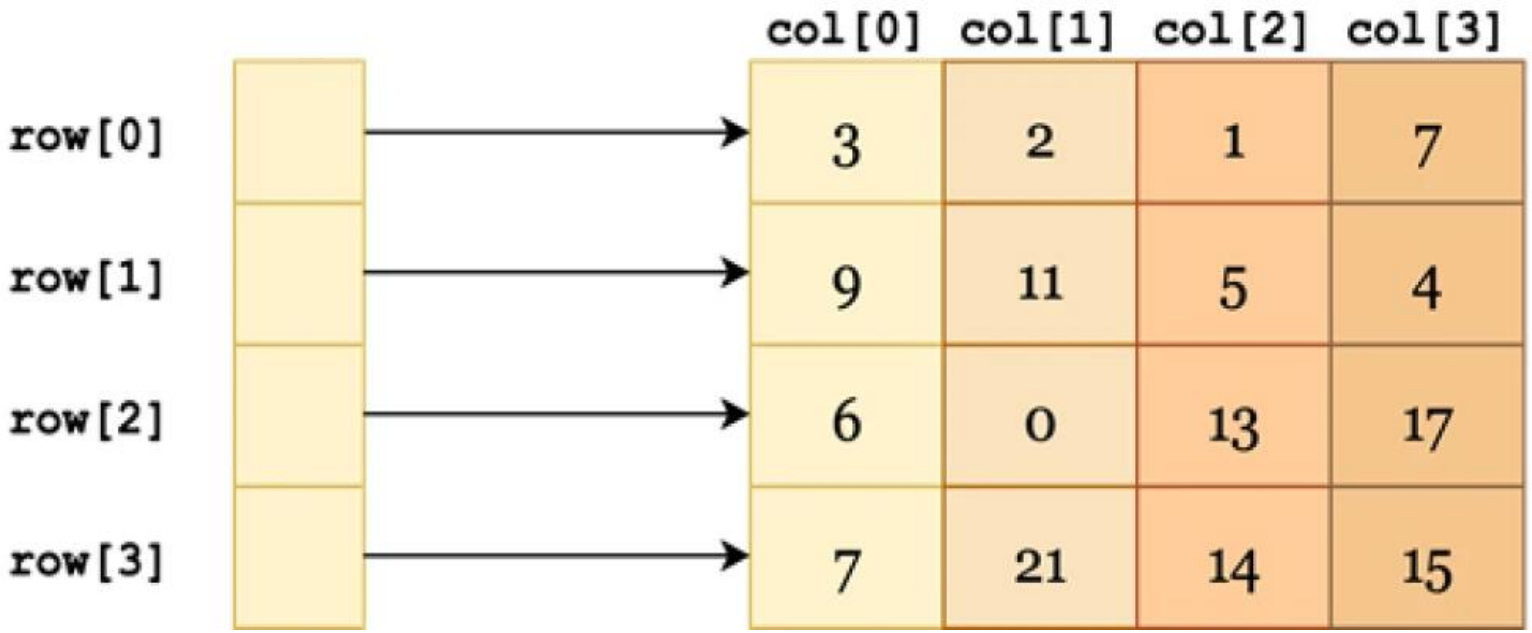
Arrays

Arrays durchlaufen:

```
for (int i = 0; i < arrayName.length; i++) {
    System.out.println(arrayName[i]);
}
```

```
for (int element : arrayName) {
    System.out.println(element);
}
```

2D-Arrays


col[0]
col[1]
col[2]
col[3]
row[0]
row[1]
row[2]
row[3]
3
2
1
7
9
11
5
4
6
0
13
17
7
21
14
15

2D-Arrays

- Deklaration & Initialisierung: `int[][] matrix = new int[3][3];`
- Zurgiff auf die Elemente: matrix[Zeile][Spalte]

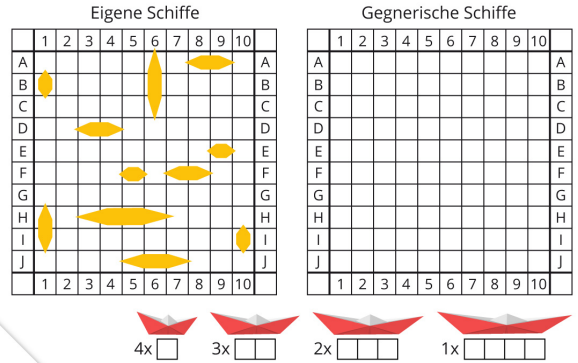
Aufgabe in Intellij

Array-Klasse

- Sammlung statischer Methoden zur Manipulation von Arrays
- Wichtigste Methoden:
  - Arrays.sort(arrayName): Sortiert arrayName in aufsteigender Reihenfolge
  - Arrays.binarySearch(arrayName, key): Sucht nach einem bestimmten Wert und gibt den Index zurück
  - Arrays.fill(arrayName, value): Befüllt ein Array oder einen Teil eines Arrays mit bestimmten Werten
  - Arrays.equals(array1, array2): Vergleicht die Gleichheit zweier Arrays
  - Arrays.copyOf(arrayName, newLength): Erstellt eine Kopie mit neuer Länge
  - Arrays.toString(arrayName): Gibt eine String-Darstellung des Arrays zurück
  - Mehr unter: https://docs.oracle.com/javase/8/docs/api/?java/util/Arrays.html

Programmier-Aufgabe

Schiffe versenken II



Talu

1. Erstelle eine Klasse *Board*.
2. Diese Klasse soll im Konstruktor ein Spielbrett konstruieren, welches aus einem 2D-Array besteht.
   - Das Brett soll variabel und quadratisch sein, aber mindestens 3x3.
   - Das Spielfeld soll im Vorfeld mit Wasser befüllt werden.

STOP

10 min

3. Zusätzlich brauchen wir noch folgende Methoden:
   - Zum platzieren von Schiffen (**Beachte**: Wie beuge ich vor, dass Schiffe über den Spielfeldrand kommen?)
   - Zum Schießen und Markieren von Hits/Misses (**Tipp**: Hierfür benötigt *Ship* auch noch eine zusätzliche Methode)
   - Zum darstellen in der Konsole des Spielfelds
   - Zum Hinzufügen eines Schiffes (**Tipp**: Es wird ein eigener Array *Ships* benötigt.)

30 min

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