**Important Java Fundamentals Concepts**

**Data Types / Типове данни**

**Integer Data Types / Целочислени типове данни**

1. byte - Byte.parseByte(scanner.nextLine());

2. short - Short.parseShort(scanner.nextLine());

3. int - Integer.parseInt(scanner.nextLine());

4. long - Long.parseLong(scanner.nextLine());

5. Big Integer - new Big Integer(scanner.nextLine());

**Decimal Data Types / Дробни типове данни**

1. float - Float.parseFloat(scanner.nextLine());

2. double - Double.parseDouble(scanner.nextLine());

**Boolean Data Type / Булев тип данни**

1. boolean - true / false или булев израз, в който има оператор за сравнение

**Character Data Type / Символен тип данни**

1. char - scanner.nextLine().charAt(0);

**Преобразуване между типовете**

Кастване

char symbol = 'A';

int asciiValue1 = (int) symbol; // explicit casting, символ към аски код

int asciiValue2 = symbol; // implicit casting

String number = "145";

char symbol = number.charAt(0); //от стринг към символ

String symbolAsText = symbol + ""; // от символ към стринг

int digit = Integer.parseInt(symbolAsText); // от стринг към int

int digit = Integer.parseInt(inputNumber.charAt(position) + "") //позиция от стринг към инт

**Arrays / Масиви**

**Четене на масив от конзолата**

1. Масив от текстове: String[] stringArr = scanner.nextLine().split(" ");

2. Масив от цели числа: int[] integerArr = Arrays.stream(scanner.nextLine().split(" ")).mapToInt(Integer::parseInt).toArray();

3. Масив от дробни числа: double[] doubleArr = Arrays.stream(scanner.nextLine().split(" ")).mapToDouble(Double::parseDouble).toArray();

4. Масив от символи: char[] charArr = scanner.nextLine().toCharArray();

**Достъпване на елемент на масив**

1. Масив от текстове: String firstItem = stringArr[0];

2. Масив от цели числа: int firstItem = integerArr[0];

3. Масив от дробни числа: double firstItem = doubleArr[0];

4. Масив от символи: char firstItem = charArr[0];

**### Сортиране на масив**

1. Масив от текстове: Arrays.sort(stringArr);

2. Масив от цели числа: Arrays.sort(integerArr);

3. Масив от дробни числа: Arrays.sort(doubleArr);

4. Масив от символи: Arrays.sort(charArr);

**Отпечатване на масив**

1. For цикъл:

for (int i = 0; i < integerArr.length; i++) {

System.out.print(integerArr[i] + " ");

}

2. Foreach цикъл:

for (int item : integerArr) {

System.out.print(item + " ");

}

3. Метод String.join\*\* (само за масив от текстове): \*\*System.out.println(String.join(", ", stringArr));

4. Метод replaceAll: System.out.println(Arrays.toString(integerArr).replaceAll("[\\[\\]]",""));

**Lists / Листове**

**Четене на лист от конзолата**

1. Лист от текстове: List<String> stringList = Аrrays.stream(scanner.nextLine().split(" ")).collect(Collectors.toList());

2. Лист от цели числа: List<Integer> integerList = Arrays.stream(scanner.nextLine().split(" ")).map(Integer::parseInt).collect(Collectors.toList());

3. Лист от дробни числа: List<Double> doubleList = Arrays.stream(scanner.nextLine().split(" ")).map(Double::parseDouble).collect(Collectors.toList());

**Методи**

1. integerList.get(index) – връща елемента на позицията (index), подадена като аргумент в скобите

2. integerList.size() – връща цяло число – брой на елементите в листа

3. integerList.indexOf(element) – връща цяло число – позицията, на която се намира елемента (element)

4. integerList.isEmpty() – връща булева стойност true при празен лист, с 0 елемента; връша булева стойност false при пълен лист, с минимум 1 елемент в него

5. integerList.contains(element) – връща булева стойност true, ако елемента се съдържа в листа; връща булева стойност false, ако елемента НЕ се съдържа в листа

6. integerList.add(element) – добавя елемента в края на листа

7. integerList.add(index, element) – вмъква елемента на позицията, която сме подали като аргумент

8. integerList.set(index, element) – заменя елемент на конкретна позиция с новия елемент

9. integerList.remove(index) – премахва елемент на дадената позиция. Не работи докато се обхожда списъка

10. integerList.remove(Integer.valueOf(elment)) – премахва елемент, ако съществува в листа

- Ако елементът е примитивен тип данни (double, int), чрез ValueOf() трябва да го превърнем в

референтен тип данни, за да знае компилатора, че това е елемент.

- Ако го оставим int компилатора ще припознае числото като индекс / позиция, а не като елемент от листа.

- Ако елемента не съществува няма да ни даде грешка, просто ще игнорира командата.

11. integerList.addAll(numList) – може да добавим всички елементи от една колекция в друга; добавят се в края на колекцията

12. Collections.sort(integerList) - сортираме елементите в листа в нарастващ ред (ascending order)

13. Collections.reverse(integerList) – обръщаме реда на елементите в листа, независимо от това дали са сортирани или не

**Отпечатване на лист**

1. For цикъл:

for (int i = 0; i < integerList.size(); i++) {

System.out.print(integerList.get(i) + " ");

}

2. Foreach цикъл:

for (int item : integerList) {

System.out.print(item + " ");

}

3. Метод String.join (само за лист от текстове): System.out.println(String.join(", ", stringList));

4. Метод replaceAll: System.out.println(integerList.toString().replaceAll("[\\[\\],]", ""));