

## Plan

# 2023-11-23

1. Введение
2. Конвейерные методы
3. Терминальные методы
4. Важные выводы

## Theory

<https://raw.githubusercontent.com/ait-tr/cohort34.2/main/basicprogramming/lesson53/resources/53.StreamAPIStream,primitivestreams.IntermediateandTerminaloperations.pptm>

## Homework

```
// 1. Подсчитать количество элементов, содержащих подстроку "10"  
// 2. Отсортировать лист в алфавитном порядке  
// 3. Отсортировать лист в математическом порядке по возрастанию  
// 4. Найти максимальное число
```

## Code

src/homework/ArraysExample.java

```
package homework;  
  
public class ArraysExample {  
  
    public static void main(String[] args) {  
  
        int[][] array = {{1, 2, 3},  
                          {4, 5, 6},  
                          {7, 8, 9}};  
  
        for (int[] row : array) {  
            for (int cell : row) {  
                System.out.print(cell + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

src/homework/SupplierTest.java

```
package homework;  
  
import java.util.Random;  
import java.util.function.Supplier;  
  
public class SupplierTest {  
  
    private static int counter;  
    private static String[] players = {"Jack", "John", "Bob", "Ben", "Zack", "Greg", "Steve"};
```

```
public static void main(String[] args) {

    Supplier<Integer> idGenerator = () -> ++counter;

    for (int i = 0; i < 5; i++) {
        System.out.println("Сгенерированный идентификатор - " + idGenerator.get());
    }

    Supplier<Integer> randomGenerator = () -> new Random().nextInt(11) + 10;

    for (int i = 0; i < 10; i++) {
        System.out.println("Случайное значение от 10 до 20 - " + randomGenerator.get());
    }

    Supplier<String> nameGenerator = () -> players[new Random().nextInt(players.length)];

    for (int i = 0; i < 5; i++) {
        System.out.println(nameGenerator.get() + ", it's your turn now!");
    }

    Supplier<Integer> supplier = () -> 7;
}
}
```

src/task\_01/Main.java

```
package task_01;

import java.util.HashSet;
import java.util.List;
import java.util.Set;
import java.util.stream.Collectors;

public class Main {

    public static void main(String[] args) {

        List<Integer> numbers = List.of(2, 7, 1, 8, 0, 11, 14, 9, 7);

        // Получить из листа коллекцию нечётных чисел без дубликатов.

        String value = "Hello!";
        // 1 способ - при помощи цикла
        Set<Integer> set = new HashSet<>();
        for (Integer current : numbers) {
            if (current % 2 != 0) {
                set.add(current);
                System.out.println("В сет добавлен элемент " + current);
            }
        }

        System.out.println("Исходный лист - " + numbers);
        System.out.println("Получившийся сет - " + set);

        // 2 способ - при помощи стрима

        Set<Integer> result = numbers.stream() // стрим 2, 7, 1, 8, 0, 11, 14, 9, 7
            .filter(x -> x % 2 != 0) // стрим 7, 1, 11, 9, 7
            .collect(Collectors.toSet()); // сет 7, 1, 11, 9
    }
}
```

```
System.out.println("Результат - " + result);

// Получить сумму всех нечётных элементов листа

// 1 способ - при помощи цикла
int sum = 0;
for (Integer current : numbers) {
    if (current % 2 != 0) {
        sum += current;
    }
}
System.out.println("Сумма нечётных чисел - " + sum);

// 2 способ - при помощи стрима
int sum1 = numbers.stream()// стрим 2, 7, 1, 8, 0, 11, 14, 9, 7
    .filter(x -> x % 2 != 0)// стрим 7, 1, 11, 9, 7
    .reduce((x, y) -> x + y)
    // 7, 1, 11, 9, 7 -> 8, 11, 9, 7 -> 19, 9, 7 -> 28, 7 -> 35
    .orElse(-1);

System.out.println("Сумма нечётных чисел - " + sum1);
}
}
```

src/task\_02/Main.java

```
package task_02;

import java.util.List;

public class Main {

    public static void main(String[] args) {

        List<String> numbers = List.of("10", "20", "30", "10", "100", "50");

        // 10, 10, 100, 20, 30, 50 - алфавитная сортировка
        // 10, 10, 20, 30, 50, 100 - математическая сортировка по возрастанию

        String result = numbers.stream()
            .reduce((x, y) -> x + y)
            .orElse("");

        System.out.println("Результат - " + result);

        // Получить сумму чисел математически
        // Преобразовать строки в числа - сложить их математически -
        // результат преобразовать в стринг

        String result1 = String.valueOf(
            numbers.stream() // стрим строк "10", "20", "30", "10", "100", "50"
                .map(x -> Integer.parseInt(x)) // стрим чисел 10, 20, 30, 10, 100, 50
                .reduce((x, y) -> x + y) // 220
                .orElse(0)
        );

        System.out.println("Результат - " + result1);
    }
}
```

```

// Найти количество элементов, которые меньше 50

long result2 = numbers.stream() // стрим строк "10", "20", "30", "10", "100", "50"
    .map(x -> Integer.parseInt(x)) // стрим чисел 10, 20, 30, 10, 100, 50
    .filter(x -> x < 50) // стрим чисел 10, 20, 30, 10
    .count(); // 4

System.out.println("Результат - " + result2);

// 1. Подсчитать количество элементов, содержащих подстроку "10"
// 2. Отсортировать лист в алфавитном порядке
// 3. Отсортировать лист в математическом порядке по возрастанию
// 4. Найти максимальное число
}
}

```

src/task\_03/Main.java

```

package task_03;

import java.util.List;
import java.util.stream.Collectors;

public class Main {

    public static void main(String[] args) {

        List<String> list1 = List.of("A", "B", "C");
        List<String> list2 = List.of("D", "E", "F");
        List<String> list3 = List.of("G", "H", "I");

        List<List<String>> listOfLists = List.of(list1, list2, list3);

        // Преобразовать лист листов в простой лист, содержащий все элементы

        // "A", "B", "C"
        // "D", "E", "F"    ->  "A", "B", "C", "D", "E", "F", "G", "H", "I"
        // "G", "H", "I"

        System.out.println("На входе:");
        listOfLists.forEach(x -> System.out.println(x));

        List<String> result = listOfLists.stream() // стрим листов
            .flatMap(x -> x.stream()) // стрим стрингов, объединённый из трёх стримов стрингов
            .collect(Collectors.toList());

        System.out.println("На выходе:");
        System.out.println(result);
    }
}

```