|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/12 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений.**

**Отчет**

**по лабораторной работе № 9**

**Вариант № 3**

**Название:** Stream API

**Дисциплина:** Языки программирования для работы с большими данными

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-23М |  |  | В.Н. Зыкин |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Цель:** изучить и освоить принципы работы с Stream API в Java.

**Задание 1:** Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно. Задана коллекция чисел. Вернуть сумму нечетных чисел.

Код класса Main:

import java.util.List;  
import java.util.Scanner;  
import java.util.stream.Collectors;  
import java.util.stream.Stream;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
 System.out.print("Введите числа через пробел: ");  
 String input = scanner.nextLine();  
  
 List<Integer> numbers = parseInput(input);  
 int sumOfOddNumbers = numbers.stream()  
 .filter(n -> n % 2 != 0) // Фильтрация нечетных чисел  
 .mapToInt(Integer::intValue) // Преобразование в IntStream  
 .sum(); // Нахождение суммы  
  
 System.out.println("Сумма нечетных чисел: " + sumOfOddNumbers);  
 }  
  
 private static List<Integer> parseInput(String input) {  
 // Разбиение входной строки на числа  
 return Stream.of(input.split("\\s+"))  
 .map(Integer::parseInt) // Преобразование строк в целые числа  
 .collect(Collectors.toList());  
 }  
}

Результат работы программы представлен на рисунке 1.

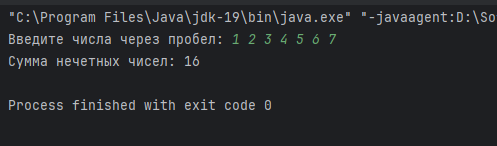


Рисунок 1 – Результат работы программы

**Задание 2:** Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно. Задана коллекция чисел. Разделить числа на четные и нечетные.

Код класса Main:

import java.util.List;  
import java.util.Scanner;  
import java.util.stream.Collectors;  
import java.util.stream.Stream;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
 System.out.print("Введите числа через пробел: ");  
 String input = scanner.nextLine();  
  
 List<Integer> numbers = parseInput(input);  
  
 List<Integer> evenNumbers = numbers.stream()  
 .filter(n -> n % 2 == 0)  
 .toList();  
  
 List<Integer> oddNumbers = numbers.stream()  
 .filter(n -> n % 2 != 0)  
 .toList();  
  
 System.out.println("Четные числа: " + evenNumbers);  
 System.out.println("Нечетные числа: " + oddNumbers);  
 }  
  
 private static List<Integer> parseInput(String input) {  
 // Разбиение входной строки на числа  
 return Stream.of(input.split("\\s+"))  
 .map(Integer::parseInt) // Преобразование строк в целые числа  
 .collect(Collectors.toList());  
 }  
}

Результат работы программы показан на рисунке 2.

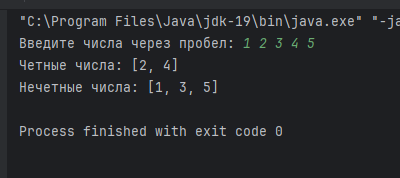


Рисунок 2 – Результат работы программы

**Задание 3:** Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно.

Задана коллекция:

(Класс People: имя и возраст)  
Collection<People> peoples = Arrays.*asList*(  
 **new** People(**"Ivan"**, 16),  
 **new** People(**"Petr"**, 23),  
 **new** People(**"Maria"**, 42)  
);

Отсортировать по возрасту.

Код класса Main:

import java.util.Arrays;  
import java.util.Collection;  
import java.util.Comparator;  
import java.util.List;  
  
public class Main {  
 static class People {  
 private final String name;  
 private final int age;  
  
 public People(String name, int age) {  
 this.name = name;  
 this.age = age;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public int getAge() {  
 return age;  
 }  
 }  
  
 public static void main(String[] args) {  
 Collection<People> peoples = Arrays.asList(  
 new People("Ivan", 16),  
 new People("Petr", 23),  
 new People("Maria", 42)  
 );  
  
 List<People> sortedByAge = peoples.stream()  
 .sorted(Comparator.comparingInt(People::getAge))  
 .toList();  
  
 sortedByAge.forEach(p -> System.out.println(p.getName() + ": " + p.getAge()));  
 }  
}

Результат работы программы представлен на рисунке 3.

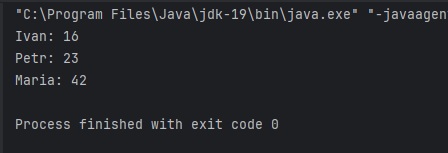


Рисунок 3 – Результат работы программы

**Задание 4:** Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно. Задана коллекция строк. Преобразовать в MAP, где первый символ ключ, второй – значение.

Код класса Main:

import java.util.HashMap;  
import java.util.Map;  
import java.util.Scanner;  
import java.util.stream.Collectors;  
import java.util.stream.Stream;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
 System.out.print("Введите строки через пробел: ");  
 String input = scanner.nextLine();  
  
 Map<Character, Character> resultMap = Stream.of(input.split("\\s+")) // Разбиение входной строки на слова  
 .map(String::toCharArray) // Преобразование каждого слова в массив символов  
 .flatMap(chars -> chars.length >= 2 ? Stream.of(chars) : Stream.empty()) // Преобразование массивов символов в поток символов  
 .collect(Collectors.toMap(chars -> chars[0], chars -> chars[1], (existing, replacement) -> replacement, HashMap::new)); // Создание Map, где первый символ - ключ, второй - значение  
  
 System.out.println("Результат: " + resultMap);  
 }  
}

Результат работы программы представлен на рисунке 4.

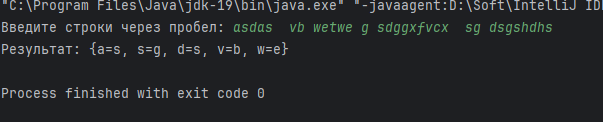


Рисунок 4 – Результат работы программы

**Вывод:** были освоены принципы работы со Stream APв Java.