*Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«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

ДИСЦИПЛИНА Языки программирования для работы с большими данными

**Отчет**

**по лабораторной работе № 9**

**Вариант 5**

Студент гр. ИУ6-21М **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_**Дейкин И. И.\_\_\_

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2020

**ЗАДАНИЕ 1**

*Условие*

Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно.

Задана коллекция строк. Вернуть список из коллекции без повторов.

*Решение*

Код решения приведен ниже:

|  |
| --- |
| package com.company;  import java.io.IOException;  import java.nio.file.Files;  import java.nio.file.Path;  import java.nio.file.Paths;  import java.util.ArrayList;  import java.util.List;  import java.util.stream.Collectors;  import java.util.stream.Stream;  public class Main {  public static void main(String[] args) {  List<String> lines = new ArrayList<>();  Path path = Paths.get("Input.txt");  // Читаем коллекцию из файла в lines  try (Stream<String> lineStream = Files.lines(path)) {  lines = lineStream.collect(Collectors.toList());  } catch (IOException ignored) {  }  // Выводим каждый уникальный элемент lines  lines.stream().distinct().forEach(x -> System.out.println(x));  }  } |

**ЗАДАНИЕ 2**

*Условие*

Использовать ТОЛЬКО методы Stream API. Циклов и условий быть не должно. Задана коллекция строк. Преобразовать в MAP, сгруппировав по первому символу строки.

*Решение*

Код решения приведен ниже:

|  |
| --- |
| package com.company;  import java.io.IOException;  import java.nio.file.Files;  import java.nio.file.Path;  import java.nio.file.Paths;  import java.util.ArrayList;  import java.util.List;  import java.util.Map;  import java.util.stream.Collectors;  import java.util.stream.Stream;  public class Main {  public static void main(String[] args) {  // write your code here  List<String> lines = new ArrayList<>();  Path path = Paths.get("Input.txt");  // Читаем коллекцию из файла в lines  try (Stream<String> lineStream = Files.lines(path)) {  lines = lineStream.collect(Collectors.toList());  } catch (IOException ignored) {  }  // Группируем в мап по первым символам  Map<Character, List<String>> map = lines.stream().collect(Collectors.groupingBy(s -> s.charAt(0)));  // Выводим  map.forEach((key, val) -> System.out.println(key + ": " + val));  }  } |

**ВЫВОД**

В ходе выполнения данной работы были рассмотрены инструменты Stream API, позволяющие формировать и обрабатывать потоки данных. Была произведена работа Stream API совместно с коллекциями, также проводилось группирование данных.