|  |  |
| --- | --- |
| **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\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа,**

**обработки и интерпретации больших данных.**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 6 |

**Название:**

Коллекции

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22М |  |  | Е.Е. Шморгун |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Вариант 1 Задача 10.** Определить множество на основе множества целых чисел. Создать методы для определения пересечения и объединения множеств.

Main.java

import java.util.HashSet;

import java.util.Set;

public class Main {

public static void main(String[] args) {

Set<Integer> set1 = new HashSet<>();

Set<Integer> set2 = new HashSet<>();

set1.add(1); set1.add(2); set1.add(3);

set2.add(1); set2.add(2); set2.add(4);

System.out.printf("Set intersection: %s\n", set\_intersection(set1, set2));

System.out.printf("Set union: %s\n", set\_union(set1, set2));

}

public static Set<Integer> set\_intersection(Set<Integer> a, Set<Integer> b) {

Set res = new HashSet();

a.forEach(e -> {

if (b.contains(e)) {

res.add(e);

}

});

b.forEach(e -> {

if (a.contains(e)) {

res.add(e);

}

});

return res;

}

public static Set<Integer> set\_union(Set<Integer> a, Set<Integer> b) {

Set res = new HashSet();

a.forEach(e ->res.add(e));

b.forEach(e -> res.add(e));

return res;

}

}

**Вариант 1 Задача 9.** Задан файл с текстом на английском языке. Выделить все различные слова. Слова, отличающиеся только регистром букв, считать одинаковыми. Использовать класс HashSet.

Main.java

import java.io.BufferedReader;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

import java.util.stream.Collectors;

public class Main {

public static void main(String[] args) {

Path input\_path = Paths.get("input.txt");

String text = "";

try {

BufferedReader reader = Files.newBufferedReader(input\_path);

while (reader.ready()) {

text += reader.readLine();

}

} catch (IOException e) {

System.out.println(e);

return;

}

Set<String> words = Arrays

.stream(text.split("[ .,:\\n]"))

.map(word -> word.toLowerCase())

.collect(Collectors.toSet());

System.out.println(words);

}

}

**Вариант 2 Задача 9.** Дана матрица из целых чисел. Найти в ней прямоугольную подматрицу, состоящую из максимального количества одинаковых элементов. Использовать класс Stack.

Main.java

import java.util.Stack;

public class Main {

public static void main(String[] args) {

Stack<Integer> sizes\_stack = new Stack<>();

int[][] matrix = {

{ 1, 1, 2, 2, 2 }, // 0

{ 1, 1, 2, 2, 2 }, // 1

{ 3, 3, 2, 2, 2 }, // 2

{ 4, 6, 2, 2, 2 }, // 3

{ 4, 6, 6, 6, 6 }, // 3

{ 4, 6, 6, 6, 6 }, // 3

};

int n = matrix.length;

int m = matrix[0].length;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= m; j++) {

if (isSectorPresent(matrix, i, j)) {

int size = i \* j;

if (sizes\_stack.empty() || sizes\_stack.lastElement() < size) {

sizes\_stack.push(size);

}

}

}

}

System.out.println(sizes\_stack.pop());

}

public static boolean isSectorPresent(int[][] matrix, int rows, int cols) {

int n = matrix.length;

int m = matrix[0].length;

for (int i = 0; i <= n - rows; i++) {

for (int j = 0; j <= m - cols; j++) {

boolean needed\_sector = true;

int sector\_val = matrix[i][j];

for (int k = 0; k < rows; k++) {

for (int l = 0; l < cols; l++) {

if (matrix[i + k][j + l] != sector\_val) {

needed\_sector = false;

break;

}

}

if (!needed\_sector) break;

}

if (needed\_sector) return true;

}

}

return false;

}

}

**Вариант 2 Задача 10.** На прямой гоночной трассе стоит N автомобилей, для каждого из которых известны начальное положение и скорость. Определить, сколько произойдет обгонов.

Main.java

public class Main {

public static void main(String[] args) {

double[][] cars = {

{ 1, 10 },

{ 2, 15 },

{ 3, 9 },

{ 4, 16 },

{ 5, 11 }

};

int N = cars.length;

int result = 0;

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++) {

if (i == j) continue;

if (cars[i][0] < cars[j][0] && cars[i][1] > cars[j][1]) {

result += 1;

}

}

}

System.out.println(result);

}

}

**Вывод:** В данной работе я познакомился с основными коллекциями в языке Java. Узнал о принципах работы с ними и сделал несколько проектов на их основе.