|  |  |
| --- | --- |
| 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 **09.04.01/07 Интеллектуальные системы анализа, обработки и интерпретации больших данных**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 2 |

**Название:** Арифметические операции

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

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

Москва, 2024

**Цель работы:** освоить базовые принципы работы с арифметическими операциями в Java.

**Вариант: 2.**

**Задание 1:** Ввести n строк с консоли. Упорядочить и вывести строки в порядке возрастания (убывания) значений их длины.

Код решения приведен в листингах 1 и 2.

Листинг 1 — Задание 1 (1)

import java.util.ArrayList;  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
  
 System.*out*.println("Введите количество строк");  
  
 int count = 0;  
 if (sc.hasNextInt()) {  
 count = sc.nextInt();  
 }  
  
 sc.nextLine();  
  
 System.*out*.println("Введите строки");  
  
 ArrayList<String> strings = new ArrayList<>();  
  
 for (int i = 0; i < count; i++) {  
 strings.add(sc.nextLine());  
 }  
  
 sc.close();  
  
 strings.sort(new CustomStringComparator());  
  
 System.*out*.println("Отсортированные строки:");  
  
 for (String string : strings) {  
 System.*out*.println(string);  
 }  
 }  
}

Листинг 2 – Задание 1 (2)

import java.util.Comparator;  
  
public class CustomStringComparator implements Comparator<String> {  
 public int compare(String o1, String o2) {  
 return Integer.*compare*(o1.length(), o2.length());  
 }  
}

**Задание 2:** Ввести n строк с консоли. Вывести на консоль те строки, длина которых меньше (больше) средней, а также длину.

Код решения приведен в листинге 3.

Листинг 3 — Задание 2

import java.util.ArrayList;  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
  
 System.*out*.println("Введите количество строк");  
  
 int count = 0;  
 if (sc.hasNextInt()) {  
 count = sc.nextInt();  
 }  
  
 sc.nextLine();  
  
 System.*out*.println("Введите строки");  
  
 ArrayList<String> strings = new ArrayList<>();  
  
 for (int i = 0; i < count; i++) {  
 strings.add(sc.nextLine());  
 }  
  
 sc.close();  
  
 var averageLength = *GetAverageLength*(strings);  
 strings.removeIf(s -> s.length() <= averageLength);  
  
 System.*out*.println("Отфильтрованные строки (" + averageLength + "):");  
  
 for (String string : strings) {  
 System.*out*.println(string + " (" + string.length() + ")");  
 }  
 }  
  
 private static int GetAverageLength(ArrayList<String> strings) {  
 var sumLength = 0;  
  
 for (String string : strings) {  
 sumLength += string.length();  
 }  
  
 return sumLength / strings.size();  
 }  
}

**Задание 3:** Выполнить циклический сдвиг заданной матрицы на k позиций вправо (влево, вверх, вниз).

Код решения приведен в листинге 4.

Листинг 4 – Задание 3

import java.util.Scanner;  
import java.util.Random;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.in);  
  
 System.out.println("Введите размерность матрицы");  
  
 int matrixSize = 0;  
 if (sc.hasNextInt()) {  
 matrixSize = sc.nextInt();  
 }  
  
 sc.nextLine();  
  
 System.out.println("Введите величину сдвига");  
  
 int offset = 0;  
 if (sc.hasNextInt()) {  
 offset = sc.nextInt();  
 }  
  
 sc.nextLine();  
  
 sc.close();  
  
 var matrix = new int[matrixSize][matrixSize];  
  
 Random random = new Random();  
 for (int i = 0; i < matrixSize; i++) {  
 for (int j = 0; j < matrixSize; j++) {  
 matrix[i][j] = random.nextInt(-matrixSize, matrixSize + 1);  
 }  
 }  
  
 System.out.println("\nИсходная матрица:");  
 PrintMatrix(matrix);  
  
 var newMatrix = new int[matrixSize][matrixSize];  
  
 for (int i = 0; i < matrixSize; i++) {  
 for (int j = 0; j < matrixSize; j++) {  
 int newJ = j + offset;  
 int newI = i;  
  
 while (newJ >= matrixSize) {  
 newJ -= matrixSize;  
 newI = newI + 1;  
  
 if (newI == matrixSize) {  
 newI = 0;  
 }  
 }  
  
 newMatrix[newI][newJ] = matrix[i][j];  
 }  
 }  
  
 System.out.printf("\nИсходная матрица, циклически сдвинутая на %d\n", offset);  
 PrintMatrix(newMatrix);  
 }  
  
 private static void PrintMatrix(int[][] matrix) {  
 for (int[] line : matrix) {  
 for (int element : line)  
 {  
 System.out.printf("%4d", element);  
 }  
  
 System.out.println();  
 }  
 }  
}

**Задание 4:** Найти и вывести наибольшее число возрастающих (убывающих) элементов матрицы, идущих подряд.

Код решения приведен в листинге 5.

Листинг 5 — Задание 4

import java.util.Random;  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.in);  
  
 System.out.println("Введите размерность матрицы");  
  
 int matrixSize = 0;  
 if (sc.hasNextInt()) {  
 matrixSize = sc.nextInt();  
 }  
  
 sc.close();  
  
 var matrix = new int[matrixSize][matrixSize];  
  
 Random random = new Random();  
 for (int i = 0; i < matrixSize; i++) {  
 for (int j = 0; j < matrixSize; j++) {  
 matrix[i][j] = random.nextInt(-matrixSize, matrixSize + 1);  
 }  
 }  
  
 System.out.println("\nИсходная матрица:");  
 PrintMatrix(matrix);  
  
 var maxIncreasingSequenceLength = 1;  
 var currentIncreasingSequenceLength = 1;  
 int previousElement = matrix[0][0];  
  
 for (int i = 0; i < matrixSize; i++) {  
 for (int j = 0; j < matrixSize; j++) {  
 if (i == 0 && j == 0) {  
 continue;  
 }  
  
 if (matrix[i][j] > previousElement) {  
 currentIncreasingSequenceLength++;  
 } else {  
 currentIncreasingSequenceLength = 1;  
 }  
  
 if (currentIncreasingSequenceLength > maxIncreasingSequenceLength) {  
 maxIncreasingSequenceLength = currentIncreasingSequenceLength;  
 }  
  
 previousElement = matrix[i][j];  
 }  
 }  
  
 System.out.printf(  
 "\nНаибольшее число элементов, идущих подряд и образующих возрастающую последовательность: %d",  
 maxIncreasingSequenceLength);  
 }  
  
 private static void PrintMatrix(int[][] matrix) {  
 for (int[] line : matrix) {  
 for (int element : line)  
 {  
 System.out.printf("%4d", element);  
 }  
  
 System.out.println();  
 }  
 }  
}

**Вывод:** в результате выполнения лабораторной работы были разработаны несколько программ в соответствие с требованиями задания.