|  |  |
| --- | --- |
| 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**

**Вариант № 12**

**Название:** арифметические операции

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

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

Москва, 2024

**Цель:** освоить принципы арифметических операций на языке программирования Java.

**Задание 1:** ввести n строк с консоли. Вывести на консоль те строки, длина которых меньше (больше) средней, а также длину. В приведенных ниже заданиях необходимо вывести внизу фамилию разработчика, дату и время получения задания, а также дату и время сдачи задания. Для получения последней даты и времени следует использовать класс Date.

Код класса Main:

import java.util.ArrayList;

import java.util.Scanner;

import java.util.Date;

public class Main {

public static void main(String[] args) {

System.out.println("Сколько строк будет вводится?");

Scanner console = new Scanner(System.in);

int col = console.nextInt();

ArrayList<String> arrayList = new ArrayList<>();

System.out.println("Вводите строки");

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

arrayList.add(console.next());

}

float Srdlina = 0;

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

Srdlina += arrayList.get(i).length();

}

Srdlina /= col;

System.out.println("Средняя длина:" + Srdlina);

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

if (arrayList.get(i).length() < Srdlina){

System.out.println(arrayList.get(i) + "," + arrayList.get(i).length());

}

}

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

if (arrayList.get(i).length() > Srdlina){

System.out.println(arrayList.get(i) + "," + arrayList.get(i).length());

}

}

System.out.println("Kulinich");

Date now = new Date();

System.out.println("Задание получено: Fri Feb 09 14:30:00 GMT+03:00 2024");

System.out.println("Текущая дата и время: " + now);

}

}Работа программы показана на рисунке 1.

Изображение выглядит как текст, Мультимедийное программное обеспечение, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 1 – Работа программы

**Задание 2:** ввести n слов с консоли. Найти слово, в котором число различных символов минимально. Если таких слов несколько, найти первое из них.

Код класса Main:

import java.util.ArrayList;

import java.util.Date;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

System.out.println("Сколько слов будет вводится?");

Scanner console = new Scanner(System.in);

int col = console.nextInt();

ArrayList<String> arrayList = new ArrayList<>();

System.out.println("Вводите слова");

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

arrayList.add(console.next());

}

int[] HowSimvol; // объявление массива для количества уникальных символов

HowSimvol = new int[col];

int kol;

String slovo;

int pov = 0;

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

slovo = arrayList.get(i);

for (int j = 0; j < slovo.length(); j++){

for (int k = j; k >= 0; k--){

if (slovo.charAt(j) == slovo.charAt(k)){

pov++;

}

}

if (pov == 1){

HowSimvol[i]++;

}

pov = 0;

}

}

int min = 100000;

int min\_i = 0;

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

if (HowSimvol[i] < min){

min = HowSimvol[i];

min\_i = i;

}

}

System.out.println(arrayList.get(min\_i));

System.out.println("Kulinich");

Date now = new Date();

System.out.println("Задание получено: Fri Feb 09 14:30:00 GMT+03:00 2024");

System.out.println("Текущая дата и время: " + now);

}

}

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание

Рисунок 2 – Работа программы

Работа программы показана на рисунке 2.

**Задание 3:** найти и вывести наибольшее число возрастающих (убывающих) элементов матрицы, идущих подряд.

Код класса Main:

import java.util.Scanner;

import java.util.ArrayList;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Введите количество строк матрицы:");

int rows = scanner.nextInt();

System.out.println("Введите количество столбцов матрицы:");

int cols = scanner.nextInt();

double[][] matrix = new double[rows][cols];

// Заполнение матрицы случайными числами

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

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

matrix[i][j] = Math.random() \* 20.0 - 10.0;

}

}

System.out.println("Сгенерированная матрица:");

for (double[] row : matrix) {

for (double num : row) {

System.out.printf("%.2f ", num);

}

System.out.println();

}

ArrayList<Double> maxIncreasing = new ArrayList<>();

ArrayList<Double> maxDecreasing = new ArrayList<>();

for (double[] row : matrix) {

ArrayList<Double> temp = findMaxSequence(row, true);

if (temp.size() > maxIncreasing.size()) {

maxIncreasing = temp;

}

temp = findMaxSequence(row, false);

if (temp.size() > maxDecreasing.size()) {

maxDecreasing = temp;

}

}

System.out.print("Наибольшая возрастающая последовательность: ");

for (double num : maxIncreasing) {

System.out.printf("%.2f ", num);

}

System.out.println();

System.out.print("Наибольшая убывающая последовательность: ");

for (double num : maxDecreasing) {

System.out.printf("%.2f ", num);

}

System.out.println();

scanner.close();

}

private static ArrayList<Double> findMaxSequence(double[] row, boolean isIncreasing) {

ArrayList<Double> maxSequence = new ArrayList<>();

ArrayList<Double> currentSequence = new ArrayList<>();

for (int i = 0; i < row.length; i++) {

if (i == 0 || (isIncreasing && row[i] > row[i - 1]) || (!isIncreasing && row[i] < row[i - 1])) {

currentSequence.add(row[i]);

} else {

if (currentSequence.size() > maxSequence.size()) {

maxSequence = new ArrayList<>(currentSequence);

}

currentSequence.clear();

currentSequence.add(row[i]);

}

}

if (currentSequence.size() > maxSequence.size()) {

maxSequence = new ArrayList<>(currentSequence);

}

return maxSequence; // Возвращаем массив с наибольшей последовательностью

}

}Работа программы показана на рисунке 3.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – Работа программы

**Задание 4:** найти сумму элементов матрицы, расположенных между первым и вторым положительными элементами каждой строки**.**

Код модуля Main:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Введите количество строк матрицы:");

int rows = scanner.nextInt();

System.out.println("Введите количество столбцов матрицы:");

int cols = scanner.nextInt();

double[][] matrix = new double[rows][cols];

// Заполнение матрицы случайными числами

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

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

matrix[i][j] = Math.random() \* 20.0 - 10.0;

}

}

System.out.println("Сгенерированная матрица:");

for (double[] row : matrix) {

for (double num : row) {

System.out.printf("%.2f ", num);

}

System.out.println();

}

// Вычисление и вывод сумм

for (double[] row : matrix) {

double sum = findSumBetweenFirstTwoPositive(row);

System.out.printf("Сумма между первым и вторым положительным элементами: %.2f\n", sum);

}

scanner.close();

}

private static double findSumBetweenFirstTwoPositive(double[] row) {

int firstPositiveIndex = -1;

int secondPositiveIndex = -1;

for (int i = 0; i < row.length; i++) {

if (row[i] > 0) {

if (firstPositiveIndex == -1) {

firstPositiveIndex = i;

} else {

secondPositiveIndex = i;

break;

}

}

}

if (firstPositiveIndex != -1 && secondPositiveIndex != -1) {

double sum = 0;

for (int i = firstPositiveIndex + 1; i < secondPositiveIndex; i++) {

sum += row[i];

}

return sum;

}

return 0; // Если в строке меньше двух положительных элементов, сумма будет равна 0

}

}

Работа программы показана на рисунке 4.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 4 – Работа программы

**Вывод:** были освоены принципы арифметических операций на языке программирования Java.