Чернівецький національний університет імені Юрія Федьковича

Інститут фізико-технічних та комп’ютерних наук

Кафедра Математичних проблем управління і кібернетики

Лабораторна робота № 1

“Тема: Основи мови Java.”

з дисципліни “ Java”

Варіант № 18

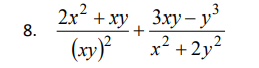
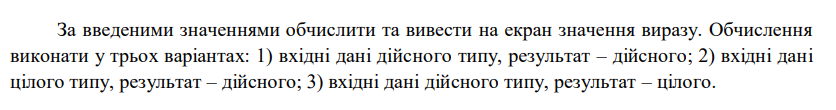
Виконав:

ст. гр. 241 Фрасинюк О. Б.

Прийняв:

Викладач Лазорик В.В.

Чернівці – 2024



**Код:**

import java.util.Scanner;

public class laba\_1\_1 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter x: ");

        double x = scanner.nextDouble();

        System.out.print("Enter y: ");

        double y = scanner.nextDouble();

        // 1. Дійсні вхідні дані, дійсний результат

        double result1 = calculateExpression(x, y);

        System.out.println("1) Float input and result: " + result1);

        // 2. Цілі вхідні дані, дійсний результат

        int intX = (int) x;

        int intY = (int) y;

        double result2 = calculateExpression(intX, intY);

        System.out.println("2) Int input float result: " + result2);

        // 3. Дійсні вхідні дані, цілий результат

        int result3 = (int) Math.round(calculateExpression(x, y));

        System.out.println("3) Float input int result: " + result3);

    }

    public static double calculateExpression(double x, double y) {

        double numerator1 = 2 \* Math.pow(x, 2) + x \* y;

        double denominator1 = Math.pow(x \* y, 2);

        double fraction1 = numerator1 / denominator1;

        double numerator2 = 3 \* x \* y - Math.pow(y, 3);

        double denominator2 = Math.pow(x, 2) + 2 \* Math.pow(y, 2);

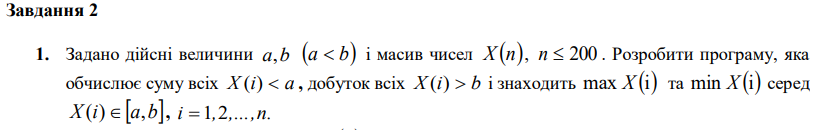
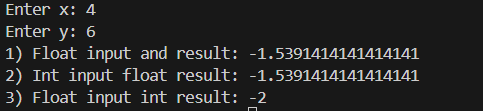
        double fraction2 = numerator2 / denominator2;

        return fraction1 + fraction2;

    }

}

**Результат:**



**Код:**

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Введення значень a, b та n

        System.out.print("Введіть значення a: ");

        double a = scanner.nextDouble();

        System.out.print("Введіть значення b (b повинно бути більше за a): ");

        double b = scanner.nextDouble();

        if (a >= b) {

            System.out.println("Помилка: b повинно бути більше за a.");

            return;

        }

        System.out.print("Введіть кількість елементів масиву n (n ≤ 200): ");

        int n = scanner.nextInt();

        if (n > 200 || n <= 0) {

            System.out.println("Помилка: n повинно бути в межах від 1 до 200.");

            return;

        }

        // Введення елементів масиву X(n)

        double[] X = new double[n];

        System.out.println("Введіть " + n + " елементів масиву:");

        for (int i = 0; i < n; i++) {

            X[i] = scanner.nextDouble();

        }

        // Ініціалізація змінних для обчислень

        double sumBelowA = 0; // Сума всіх X(i) < a

        double productAboveB = 1; // Добуток всіх X(i) > b

        boolean hasProductAboveB = false; // Прапорець для перевірки добутку

        Double maxInRange = null; // Максимум серед X(i) ∈ [a, b]

        Double minInRange = null; // Мінімум серед X(i) ∈ [a, b]

        // Обчислення

        for (double x : X) {

            if (x < a) {

                sumBelowA += x;

            } else if (x > b) {

                productAboveB \*= x;

                hasProductAboveB = true;

            } else { // x знаходиться в межах [a, b]

                if (maxInRange == null || x > maxInRange) {

                    maxInRange = x;

                }

                if (minInRange == null || x < minInRange) {

                    minInRange = x;

                }

            }

        }

        // Вивід результатів

        System.out.println("Сума всіх елементів, що менші за a: " + sumBelowA);

        if (hasProductAboveB) {

            System.out.println("Добуток всіх елементів, що більші за b: " + productAboveB);

        } else {

            System.out.println("Не знайдено елементів, більших за b.");

        }

        if (maxInRange != null && minInRange != null) {

            System.out.println("Максимум серед елементів у діапазоні [a, b]: " + maxInRange);

            System.out.println("Мінімум серед елементів у діапазоні [a, b]: " + minInRange);

        } else {

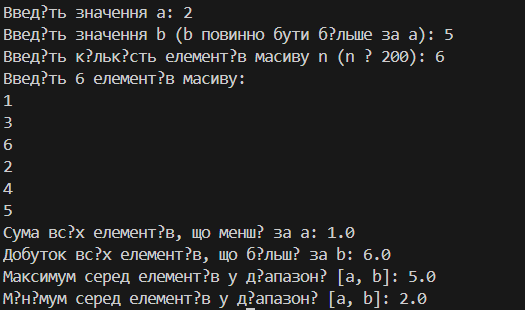
            System.out.println("Не знайдено елементів у діапазоні [a, b].");

        }

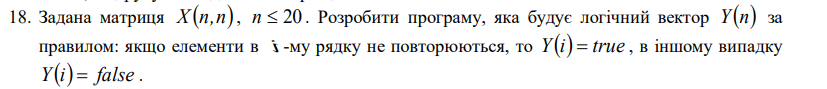
    }

}

**Результат:**



Завдання 3:



**Код:**import java.util.HashSet;

import java.util.Scanner;

public class laba\_1\_3 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Введення розміру матриці n

        System.out.print("Введіть розмір матриці n (n ≤ 20): ");

        int n = scanner.nextInt();

        // Перевірка обмеження на розмір

        if (n > 20 || n <= 0) {

            System.out.println("Помилка: n повинно бути в межах від 1 до 20.");

            return;

        }

        // Введення елементів матриці X(n, n)

        int[][] X = new int[n][n];

        System.out.println("Введіть елементи матриці " + n + " x " + n + ":");

        for (int i = 0; i < n; i++) {

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

                X[i][j] = scanner.nextInt();

            }

        }

        // Створення логічного вектора Y(n) для зберігання результату

        boolean[] Y = new boolean[n];

        // Перевірка кожного рядка на унікальність елементів

        for (int i = 0; i < n; i++) {

            // Використовуємо HashSet для перевірки унікальності елементів у рядку

            HashSet<Integer> set = new HashSet<>();

            boolean unique = true;

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

                // Якщо елемент вже присутній в HashSet, значить рядок має повторення

                if (set.contains(X[i][j])) {

                    unique = false;

                    break;

                }

                set.add(X[i][j]);

            }

            // Якщо всі елементи унікальні, то Y[i] = true, інакше Y[i] = false

            Y[i] = unique;

        }

        // Вивід результату

        System.out.println("Логічний вектор Y:");

        for (int i = 0; i < n; i++) {

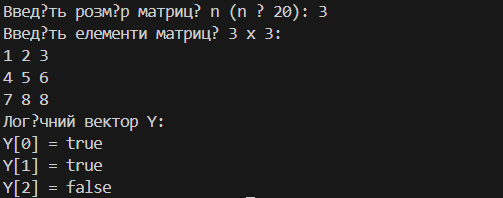
            System.out.println("Y[" + i + "] = " + Y[i]);

        }

    }

}

**Результат:**



Завдання 4:



КОД:

import java.util.Scanner;

public class laba\_1\_4s {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Введення тексту

        System.out.println("Введіть текст:");

        String inputText = scanner.nextLine();

        // Розділяємо текст на слова з урахуванням пробілів і розділових знаків

        String[] words = inputText.split("\\b");

        // Створюємо новий рядок для збереження результату

        StringBuilder result = new StringBuilder();

        // Проходимо по кожному слову

        for (String word : words) {

            if (word.matches("\\w+")) { // Перевіряємо, що це слово (не розділовий знак)

                // Залишаємо тільки перше входження першої літери

                String processedWord = removeNextOccurrences(word);

                result.append(processedWord);

            } else {

                // Додаємо розділові знаки та пробіли без змін

                result.append(word);

            }

        }

        // Вивід обробленого тексту

        System.out.println("Оброблений текст:");

        System.out.println(result.toString());

    }

    // Метод для видалення всіх наступних входжень першої літери в слові

    public static String removeNextOccurrences(String word) {

        char firstChar = word.charAt(0); // Отримуємо першу літеру

        StringBuilder processedWord = new StringBuildруддer();

        processedWord.append(firstChar); // Додаємо першу літеру в результат

        // Додаємо до результату всі інші символи, крім повторних входжень першої літери

        for (int i = 1; i < word.length(); i++) {

            char currentChar = word.charAt(i);

            if (currentChar != firstChar) {

                processedWord.append(currentChar);

            }

        }

        return processedWord.toString();

    }

}

РЕЗУЛЬТАТ:

