**Мета:** Вивчити можливості організації циклів в програмах мовою Java.

**Хід роботи**

1. У середовищі VisualCode створити проект та під'єднати його до репозиторію Git.
2. Відповідно до свого номеру у списку студентської групи обрати завдання з таблиці 1.
3. Написати відповідний вихідний код програми мовою Java.
4. За необхідності в ході виконання та обов'язково по завершенні роботи завантажити результати на репозиторій Git.
5. Створити та надати викладачеві звіт з виконання лабораторної роботи.

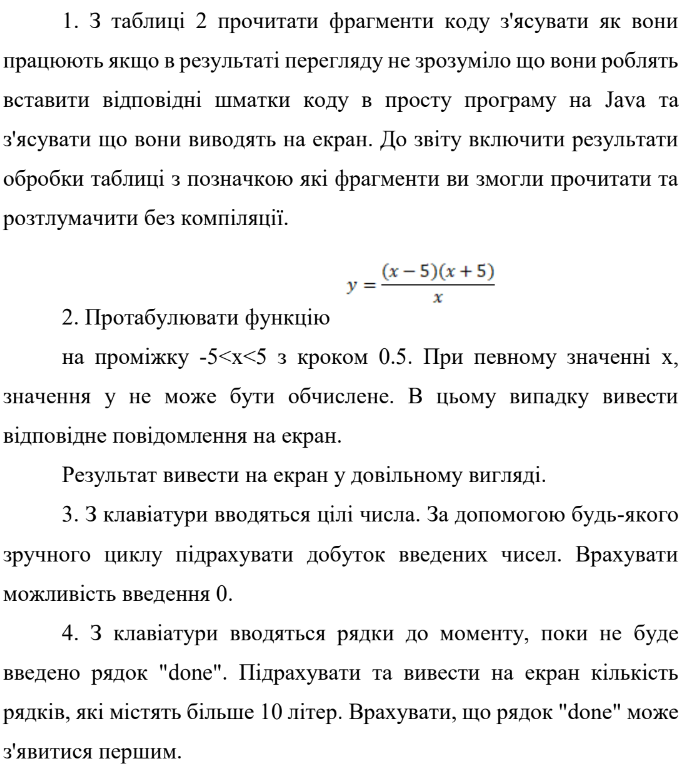


Рисунок 1 – Завдання з таблиці 1

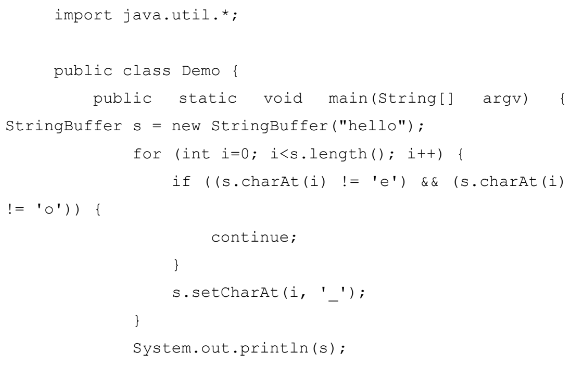


Рисунок 2 – Завдання з таблиці 2

Пояснення коду із завдання 1:

Фрагмент із класу Demo обробляє рядок hello, змінюючи всі символи, окрім 'e' і 'o', на підкреслення \_. Програма використовує цикл for, який перебирає всі символи рядка та змінює їх залежно від умови.

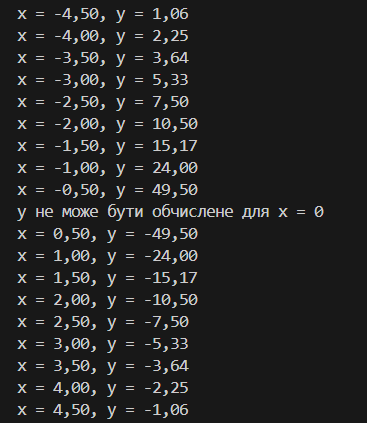


Рисунок 3 – Результат виконання завдання 2

Код завдання 2:

public class Main {

    public static void main(String[] args) {

        for (double x = -4.5; x < 5; x += 0.5) {

            if (x == 0) {

                System.out.println("y не може бути обчислене для x = 0");

            } else {

                double y = ((x - 5) \* (x + 5)) / x;

                System.out.printf("x = %.2f, y = %.2f%n", x, y);

            }

        }

    }

}

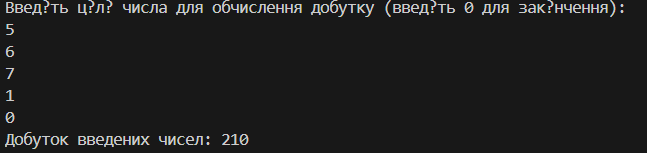


Рисунок 4 – Результат виконання завдання 3

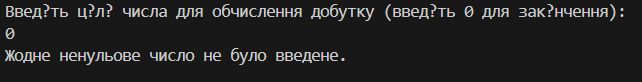


Рисунок 5 – Результат виконання завдання 3

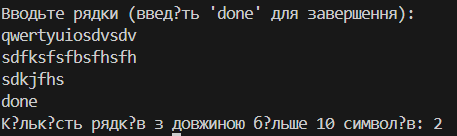


Рисунок 6 – Результат виконання завдання 4

Код для завдання 3-4:

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Завдання 3:

        long product = 1;

        boolean hasNonZero = false;

        System.out.println("Введіть цілі числа для обчислення добутку (введіть 0 для закінчення):");

        while (true) {

            int num = scanner.nextInt();

            if (num == 0) break;  // Якщо введено 0, завершуємо цикл

            product \*= num;

            hasNonZero = true;

        }

        if (hasNonZero) {

            System.out.println("Добуток введених чисел: " + product);

        } else {

            System.out.println("Жодне ненульове число не було введене.");

        }

        // Завдання 4

        int longStringCount = 0;

        System.out.println("\nВводьте рядки (введіть 'done' для завершення):");

        scanner.nextLine();

        while (true) {

            String input = scanner.nextLine();

            if (input.equals("done")) break;  // Якщо введено "done", завершуємо цикл

            if (input.length() > 10) longStringCount++;

        }

        System.out.println("Кількість рядків з довжиною більше 10 символів: " + longStringCount);

    }

}

**Контрольні питання:**

1. Де потрібно розмістити опис лічильника циклу, щоб потім скористатись ним десь нижче по коду.

Лічильник циклу потрібно оголосити поза циклом, якщо він потрібен після його завершення.

1. Який різновид циклу найбільше підходить для того, щоб протабулювати функцію.

Цикл for.

1. Які особливості і перестороги має мовна конструкція циклу while.

Цикл while може не виконатися жодного разу, якщо умова спочатку хибна, і є ризик нескінченного циклу, якщо умова завжди істинна.

1. Чим відрізняється мовна конструкція циклу while від циклу do while.

У циклі while умова перевіряється перед виконанням тіла циклу, а в do while — після, тому do while завжди виконується хоча б один раз.

1. Яке призначення мають оператори break та continue.

break завершує цикл достроково; continue переходить до наступної ітерації, пропускаючи решту коду в поточній ітерації.

1. Чи можливо використовувати у вкладених циклах різні мовні конструкції циклів.

Так, у вкладених циклах можна використовувати різні конструкції циклів, наприклад, for всередині while.

**Висновок:** вивчив можливості вводу даних з клавіатури та використання умовних конструкцій в програмах мовою Java.