

Powyższa klasa LinuxLogWriter zawiera następujące elementy:

Stałą LOG\_FILE\_PATH, która określa ścieżkę do pliku, w którym będą zapisywane logi.

Metodę main(), która tworzy instancję klasy LinuxLogWriter i wywołuje metodę run().

Metodę run(), która:

Wyświetla instrukcję dla użytkownika.

Tworzy obiekty Scanner oraz BufferedWriter do odczytu danych z klawiatury i zapisu do pliku.

Uruchamia pętlę, w której:

Odczytuje dane z klawiatury.

Sprawdza, czy użytkownik wpisał "quit" w celu zakończenia.

W przeciwnym razie wywołuje metodę writeLogEntry() w celu zapisu danych do pliku.

Obsługuje wyjątki IOException podczas zapisu do pliku.

Metodę writeLogEntry(), która:

Pobiera bieżący znacznik czasu za pomocą metody getTimestamp().

Zapisuje wpis logu w formacie "yyyy-MM-dd HH:mm:ss - <wpisany tekst>" do pliku.

Obsługuje wyjątki IOException podczas zapisu.

Metodę getTimestamp(), która zwraca bieżącą datę i godzinę w formacie "yyyy-MM-dd HH:mm:ss".

Aby użyć tej klasy, wystarczy uruchomić metodę main(). Program będzie oczekiwał na wprowadzanie danych z klawiatury, a wszystkie wpisy będą zapisywane do pliku log.txt w bieżącym katalogu. Aby zakończyć działanie programu, należy wpisać "quit".

```

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;

public class LinuxLogWriter {
    private static final String LOG_FILE_PATH = "log.txt";

    public static void main(String[] args) {
        LinuxLogWriter logWriter = new LinuxLogWriter();
        logWriter.run();
    }

    public void run() {
        System.out.println("Rozpoczynamy zapis logu do pliku log.txt. Wprowadzaj dane z klawiatury, a program będzie je zapisywał. Wpisz 'quit', aby zakończyć.");

        try (Scanner scanner = new Scanner(System.in);
            BufferedWriter writer = new BufferedWriter(new FileWriter(LOG_FILE_PATH, true))) {
            while (true) {
                String input = scanner.nextLine();
                if ("quit".equalsIgnoreCase(input)) {
                    break;
                }
                writeLogEntry(writer, input);
            }
        } catch (IOException e) {
            System.err.println("Wystąpił błąd podczas zapisu do pliku: " + e.getMessage());
        }

        System.out.println("Zapis logu zakończony.");
    }

    private void writeLogEntry(BufferedWriter writer, String logEntry) {
        try {
            String timestamp = getTimestamp();
            writer.write(timestamp + " - " + logEntry);
            writer.newLine();
            writer.flush();
        } catch (IOException e) {
            System.err.println("Wystąpił błąd podczas zapisu do pliku: " + e.getMessage());
        }
    }

    private String getTimestamp() {
        LocalDateTime now = LocalDateTime.now();
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
        return now.format(formatter);
    }
}

```