|  |  |  |  |
| --- | --- | --- | --- |
| ***SPRAWOZDANIE NR*** | | | |
| Nazwa ćwiczenia | Zapis odczyt | | Obraz zawierający tekst  Opis wygenerowany automatycznie |
| Przedmiot | Programowanie obiektowe | |
| Student  grupa | Paweł Jońca gr 7 | |
| Data ćwiczeń | 19.12.2024r | 19.12.2024r | Data oddania sprawozdania |

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, komputer, multimedia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, multimedia, oprogramowanie

Opis wygenerowany automatycznie

import javax.swing.\*; // Importowanie klas do tworzenia GUI  
import java.awt.\*; // Importowanie klas do zarządzania układem GUI  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener; // Importowanie klas do obsługi zdarzeń  
import java.io.\*; // Importowanie klas do operacji na plikach  
  
public class ZapisOdczyt {  
  
 // Stała przechowująca nazwę pliku tekstowego  
 private static final String FILE\_NAME = "plik.txt";  
  
  
 public static void main(String[] args) {  
 // Tworzenie okna aplikacji  
 JFrame frame = new JFrame("Zapis/Odczyt Pliku");  
 frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); // Zamknięcie aplikacji po zamknięciu okna  
 frame.setSize(600, 300); // Ustawienie rozmiaru okna  
 frame.setLayout(new BorderLayout()); // Ustawienie układu BorderLayout  
  
  
 // Tworzenie obszaru tekstowego z możliwością przewijania  
 JTextArea textArea = new JTextArea(); // Pole tekstowe do wprowadzania i wyświetlania tekstu  
 JScrollPane scrollPane = new JScrollPane(textArea); // Dodanie przewijania do pola tekstowego  
 frame.add(scrollPane, BorderLayout.CENTER); // Umieszczenie obszaru tekstowego w centralnej części okna  
  
  
 // Tworzenie panelu na przyciski  
 JPanel buttonPanel = new JPanel(); // Panel do przechowywania przycisków  
 frame.add(buttonPanel, BorderLayout.SOUTH); // Umieszczenie panelu w dolnej części okna  
  
 // Tworzenie przycisków  
 JButton readButton = new JButton("Odczyt z pliku"); // Przycisk do odczytu z pliku  
 JButton writeButton = new JButton("Zapis do pliku"); // Przycisk do zapisu do pliku  
 buttonPanel.add(readButton); // Dodanie przycisku odczytu do panelu  
 buttonPanel.add(writeButton); // Dodanie przycisku zapisu do panelu  
  
  
 // Obsługa przycisku "Odczyt z pliku"  
 readButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 try (BufferedReader reader = new BufferedReader(new FileReader(FILE\_NAME))) { //try-with-resources  
 textArea.setText(""); // Czyszczenie obszaru tekstowego przed odczytem  
 String line;  
 // Odczytywanie linii z pliku i dodawanie ich do obszaru tekstowego  
 while ((line = reader.readLine()) != null) {  
 textArea.append(line + "\n");  
 }  
 } catch (IOException ex) {  
 // Wyświetlenie komunikatu o błędzie, jeśli plik nie istnieje lub wystąpił inny problem  
 JOptionPane.showMessageDialog(frame,  
 "Błąd podczas odczytu pliku: " + ex.getMessage(),  
 "Błąd", JOptionPane.ERROR\_MESSAGE);  
 }  
 }  
 });  
  
  
 // Obsługa przycisku "Zapis do pliku"  
 writeButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(FILE\_NAME))) {  
 // Zapis zawartości obszaru tekstowego do pliku  
 writer.write(textArea.getText());  
 // Wyświetlenie komunikatu o sukcesie  
 JOptionPane.showMessageDialog(frame,  
 "Tekst zapisano do pliku.",  
 "Sukces", JOptionPane.INFORMATION\_MESSAGE);  
 } catch (IOException ex) {  
 // Wyświetlenie komunikatu o błędzie w przypadku problemów z zapisem  
 JOptionPane.showMessageDialog(frame,  
 "Błąd podczas zapisu pliku: " + ex.getMessage(),  
 "Błąd", JOptionPane.ERROR\_MESSAGE);  
 }  
 }  
 });  
  
 // Wyświetlenie okna aplikacji  
 frame.setVisible(true); // Ustawienie okna jako widocznego  
 }  
}

Wnioski:

W trakcie implementacji tego zadania nauczyłem się, jak tworzyć graficzny interfejs użytkownika za pomocą Swinga oraz integrować go z funkcjonalnością zapisu i odczytu plików. Poznałem również praktyczne zastosowanie mechanizmu try-with-resources, co umożliwia bezpieczne operacje na plikach bez konieczności ręcznego zamykania strumieni.