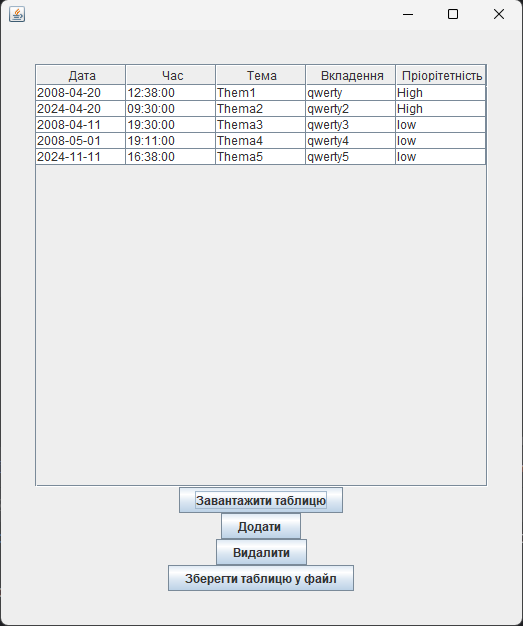
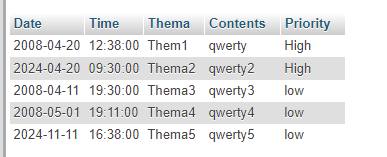
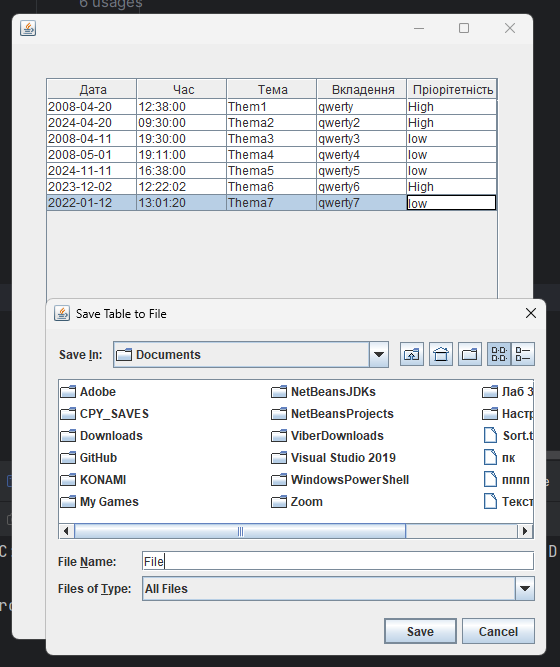
**Лабораторна робота №6(17)**

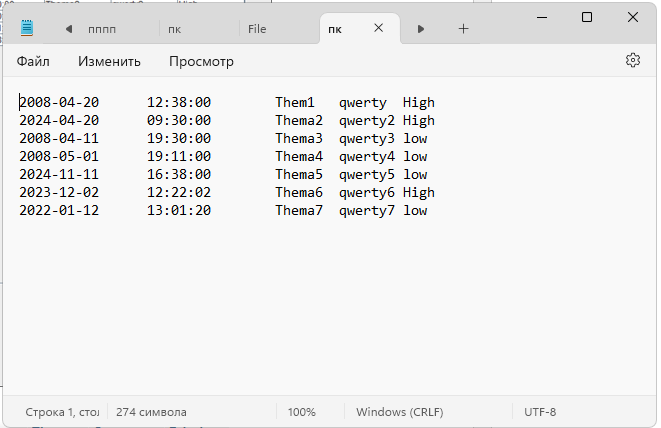
В.2

Створити програму яка виконуватиме функції персонального щоденника. (запис важливих подій за день)









import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.io.\*;  
import java.sql.\*;  
  
*/\*\*  
 \* Клас, який представляє головне вікно додатку.  
 \*/*public class NewJFrame extends javax.swing.JFrame {  
  
 */\*\*  
 \* Конструктор класу NewJFrame.  
 \*/* public NewJFrame() {  
 initComponents();  
 }  
  
 @SuppressWarnings("unchecked")  
 private void initComponents() {  
 jScrollPane1 = new javax.swing.JScrollPane();  
 jTable1 = new javax.swing.JTable();  
 jButton1 = new javax.swing.JButton();  
 jButton2 = new javax.swing.JButton();  
 jButton3 = new javax.swing.JButton();  
 jButton4 = new javax.swing.JButton();  
  
 setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);  
  
 jTable1.setModel(new javax.swing.table.DefaultTableModel(  
 new Object [][] {  
  
 },  
 new String [] {  
 "Дата", "Час", "Тема", "Вкладення", "Пріорітетність"  
 }  
 ));  
 jScrollPane1.setViewportView(jTable1);  
  
 jButton1.setText("Завантажити таблицю");  
 jButton1.addActionListener(new java.awt.event.ActionListener() {  
 public void actionPerformed(java.awt.event.ActionEvent evt) {  
 jButton1ActionPerformed(evt);  
 }  
 });  
  
 jButton2.setText("Додати ");  
 jButton2.setEnabled(false);  
 jButton2.addActionListener(new java.awt.event.ActionListener() {  
 public void actionPerformed(java.awt.event.ActionEvent evt) {  
 jButton2ActionPerformed(evt);  
 }  
 });  
  
 jButton3.setText("Видалити");  
 jButton3.setEnabled(false);  
 jButton3.addActionListener(new java.awt.event.ActionListener() {  
 public void actionPerformed(java.awt.event.ActionEvent evt) {  
 jButton3ActionPerformed(evt);  
 }  
 });  
  
 jButton4.setText("Зберегти таблицю у файл");  
 jButton4.addActionListener(new java.awt.event.ActionListener() {  
 public void actionPerformed(java.awt.event.ActionEvent evt) {  
 jButton4ActionPerformed(evt);  
 }  
 });  
  
 // Розташування компонентів в центрі вікна  
 JPanel centerPanel = new JPanel(new GridBagLayout());  
 GridBagConstraints gbc = new GridBagConstraints();  
 gbc.gridx = 0;  
 gbc.gridy = 0;  
 centerPanel.add(jScrollPane1, gbc);  
 gbc.gridy++;  
 centerPanel.add(jButton1, gbc);  
 gbc.gridy++;  
 centerPanel.add(jButton2, gbc);  
 gbc.gridy++;  
 centerPanel.add(jButton3, gbc);  
 gbc.gridy++;  
 centerPanel.add(jButton4, gbc);  
  
 getContentPane().add(centerPanel, BorderLayout.CENTER);  
  
 pack();  
 }  
  
 */\*\*  
 \* Обробник події для кнопки "Завантажити таблицю".  
 \*/* private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
 DefaultTableModel model = (DefaultTableModel) jTable1.getModel();  
 model.setRowCount(0);  
  
 try {  
 String url = "jdbc:mysql://localhost:3306/lab6";  
 String username = "root";  
 String password = "";  
 Connection connection = DriverManager.getConnection(url, username, password);  
  
 String sql = "SELECT Date, Time, Thema, Contents, Priority FROM deathnote";  
 Statement statement = connection.createStatement();  
 ResultSet resultSet = statement.executeQuery(sql);  
  
 while (resultSet.next()) {  
 Object[] rowData = {  
 resultSet.getDate("Date"),  
 resultSet.getTime("Time"),  
 resultSet.getString("Thema"),  
 resultSet.getString("Contents"),  
 resultSet.getString("Priority")  
 };  
 model.addRow(rowData);  
 }  
  
 jButton2.setEnabled(true); // Enable add button after loading data  
 jButton3.setEnabled(true); // Enable delete button after loading data  
 } catch (SQLException ex) {  
 ex.printStackTrace();  
 JOptionPane.showMessageDialog(this, "Error loading data: " + ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);  
 }  
 }  
  
 */\*\*  
 \* Обробник події для кнопки "Додати".  
 \*/* private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
 DefaultTableModel model = (DefaultTableModel) jTable1.getModel();  
 model.addRow(new Object[]{"", "", "", "", ""});  
 }  
  
 */\*\*  
 \* Обробник події для кнопки "Видалити".  
 \*/* private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
 int selectedRowIndex = jTable1.getSelectedRow();  
 if (selectedRowIndex >= 0) {  
 DefaultTableModel model = (DefaultTableModel) jTable1.getModel();  
 model.removeRow(selectedRowIndex);  
 }  
 jButton3.setEnabled(false);  
 }  
  
 */\*\*  
 \* Обробник події для кнопки "Зберегти таблицю у файл".  
 \*/* private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
 saveTableToFile(jTable1);  
 }  
  
 */\*\*  
 \* Зберігає таблицю у текстовий файл.  
 \*  
 \* @param table таблиця, яку потрібно зберегти  
 \*/* public void saveTableToFile(JTable table) {  
 DefaultTableModel model = (DefaultTableModel) table.getModel();  
 int rowCount = model.getRowCount();  
  
 JFileChooser fileChooser = new JFileChooser();  
 fileChooser.setDialogTitle("Save Table to File");  
 int userSelection = fileChooser.showSaveDialog(this);  
 if (userSelection == JFileChooser.APPROVE\_OPTION) {  
 File fileToSave = fileChooser.getSelectedFile();  
 try (PrintWriter writer = new PrintWriter(fileToSave)) {  
 for (int i = 0; i < rowCount; i++) {  
 for (int j = 0; j < model.getColumnCount(); j++) {  
 writer.print(model.getValueAt(i, j));  
 if (j < model.getColumnCount() - 1) {  
 writer.print("\t");  
 }  
 }  
 writer.println();  
 }  
 JOptionPane.showMessageDialog(this, "Table saved successfully!", "Success", JOptionPane.INFORMATION\_MESSAGE);  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 JOptionPane.showMessageDialog(this, "Error saving table: " + ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);  
 }  
 }  
 }  
  
 */\*\*  
 \* Точка входу в програму.  
 \*  
 \* @param args аргументи командного рядка  
 \*/* public static void main(String args[]) {  
 java.awt.EventQueue.invokeLater(new Runnable() {  
 public void run() {  
 new NewJFrame().setVisible(true);  
 }  
 });  
 }  
  
 // Поля класу  
 private javax.swing.JButton jButton1;  
 private javax.swing.JButton jButton2;  
 private javax.swing.JButton jButton3;  
 private javax.swing.JButton jButton4;  
 private javax.swing.JScrollPane jScrollPane1;  
 private javax.swing.JTable jTable1;  
}

**Висновки**: Цей код представляє простий додаток для управління таблицею даних, яка зберігається в базі даних MySQL. Основні функції додатку включають завантаження таблиці з бази даних, додавання нових рядків, видалення рядків та збереження таблиці у текстовий файл.

Код розділений на декілька методів для кожної функціональності, що полегшує читабельність та розуміння коду. Для кожного методу надано коментар Javadoc, який пояснює його призначення та вхідні/вихідні параметри.

Додаток також містить обробники подій для кнопок, які виконують відповідні дії, такі як завантаження, додавання, видалення та збереження таблиці. Крім того, передбачені винятки та помилки, що можуть виникнути під час виконання програми, і їх обробка за допомогою діалогових вікон повідомлень.

Цей код демонструє базовий функціонал для роботи з таблицями даних у Java Swing та взаємодії з базою даних MySQL, а також надає можливість збереження даних у текстовий файл.