МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНЙ УНІВЕРСИТЕТ

кафедра програмного забезпечення автоматизованих систем

**ЗВІТ**

**про виконання**

**практичної роботи з дисципліни**

**"Конструювання програмного забезпечення"**

|  |  |
| --- | --- |
| **Перевірив роботу:** | **Виконав роботу:** |
| доцент кафедри ПЗАС  Володимир МЕТЕЛАП | студент гр. ПЗ-2104  Олександр Ліла |

Черкаси 2023

**Практична робота 2**

**Програмне забезпечення для генерування та зберігання паролів.**

**Мета:** Створити програму-менеджер паролів на мові програмування Java, яка дозволяє користувачеві безпечно зберігати та керувати своїми паролями для різних веб-сайтів.

**Хід виконання практичної роботи:**

1. **Технічне завдання на створення менеджера паролів**

**Вимоги завдання:**

**1) Основний інтерфейс:**

Розробити програму-менеджер паролів, яка використовує графічний інтерфейс користувача.

Забезпечити основні елементи інтерфейсу, такі як поля для введення веб-сайту, імені користувача та паролю, кнопки для збереження, скасування, відображення/приховування паролю, відкриття та закриття бази даних паролів, копіювання паролю та пошуку.

**2) Функціонал збереження та відображення паролів:**

Забезпечити можливість введення та збереження даних для кожного веб-сайту (веб-сайт, ім'я користувача, пароль).

Забезпечити можливість скасування введених даних та очищення полів вводу. Реалізувати функціонал відображення або приховування паролю, щоб користувач міг перевірити правильність введеного паролю. Кнопка "Copy password" для копіювання введеного паролю у буфер обміну. Забезпечення безпечного введення та відображення паролів.

Забезпечення вибору користувачем видимості паролю для зручності введення.

**3) Функціонал відкриття та редагування бази даних:**

Збереження паролів у текстовому файлі.

Отримання можливості вибору та відкриття збереженого файлу бази даних через **JFileChooser**. Забезпечення відображення інформації про відкритий файл бази даних у заголовку вікна.

**4)** **Очищення та видалення даних:**

Додавання можливості введення веб-сайту для пошуку паролю за допомогою текстового поля та кнопки "Find".

Виведення інформації (імені користувача та паролю) для знайденого веб-сайту у відповідні поля.

**5) Пошук і відображення інформації:**

Можливість введення веб-сайту та відображення інформації (ім'я користувача та пароль) для введеного веб-сайту за допомогою кнопки "Find".

**6) Інтерфейс користувача**:

Забезпечення інтуїтивно зрозумілого та зручного інтерфейсу користувача для легкості використання програми. Використання подій та слухачів для забезпечення відповіді на дії користувача. Написання документації коду, включаючи коментарі та опис методів.

Розробка вікна "Довідка" з коротким описом функцій та можливостей програми. Забезпечення можливості відображення довідкової інформації з файлу "help.txt".

1. **Логіка роботи менеджера паролів:**

**Створення графічного інтерфейсу:**

Створення об'єктів JFrame, JPanel, та компонентів для введення (текстових полів та парольного поля), етикеток та кнопок.

Налаштування розміщення та стилізації компонентів.

Створення об'єкта PasswordManager:

**Створення об'єкта PasswordManager:** для обробки логіки збереження, відображення та копіювання паролів.

Ініціалізація компонентів для відображення паролів та файлового вибору.

**Обробка подій кнопок:** Додавання обробників подій для кожної кнопки. Виклик методів PasswordManager в залежності від подій.

**Збереження паролів:** При натисканні кнопки "Save data" викликається метод saveDataTextFields, який отримує дані від користувача та зберігає їх у текстовому файлі.

**Відображення/приховування паролів:** При натисканні кнопки "Visible password" викликається метод showVisiblePassword, який відображає або приховує символи паролю в полі введення.

**Відкриття та очищення бази даних паролів:** При натисканні кнопки "Open Database" викликається метод openDatabase або openClearDatabase для завантаження або очищення бази даних паролів.

**Копіювання паролю:** При натисканні кнопки "Copy password" викликається метод copyPasswordButton, який копіює пароль у буфер обміну.

**Пошук та відображення паролів за вказаною веб-сайтом:** При натисканні кнопки "Find" викликається метод findAndDisplayUsername, який знаходить та відображає пароль за введеним веб-сайтом, пощук відбувається в текстовому файлі, що є базою даних. Тобто в ньому зберігаються всі данні.

Загальний огляд логіки програми та взаємодії між компонентами реалізуємо у вигляді **блок-схеми** (рис.1)

Start program

Create JFrame and JPanel

Initialize PasswordManager object

Create components (text fields, labels, buttons)

Add components to the JFame

PasswordManager methods (Logic Handling)

Set up event handlers for each button

Event Handling (invoke PasswordManager methods)

Stop Program

Рисунок 1 – Блок-схема роботи менеджера паролів

**3) Ескіз менеджера паролів:**

В головному вікні програми є три тестових поля для вводу інформації для збереження:

* назва веб ресурсу;
* назва аккаунту;
* пароль.

Для збереження введених даних у головному вікні програми передбачена кнопка «Save». Під цією кнопкою розміщені дві кнопки: скидання не вірно набраних даних – «Cancel», та копіювання пароля – «Copy password».

В середній частині вікна розміщені в ряд три кнопки:

- для того щоб змінювати видимість пароля в текстовому полі;

- друга кнопка для пошуку пароля в базі даних. Пошук відбувається по назві веб-ресурса;

- третя кнопка для виведення всій бази даних в велике текстове поле внизу головного вікна програми. В це вікно виводиться вся інформація. Кожний рядок має назву веб-ресурсу, назву аккаунта та пароль.

**Закриття пароля менеджера:**

Вікно закривається за допомогою хрестика або через меню "File -> Exit".

Якщо є незбережені зміни, виводиться діалогове вікно для збереження перед виходом.

**Інтерфейс користувача:**

Інтуїтивно зрозумілі та зручні елементи для доступу до основних опцій.

Зручне розміщення елементів у вікні для ефективного використання простору. Використання стандартних графічних елементів та стилів.

**Інші аспекти:**

Забезпечення підтримки розширення можливостей менеджера паролів у майбутньому (наприклад, додаткові функції форматування, підтримка різних шифрувань даних тощо).

Website

Password

Username

Copy

Save

Cancel

Visible password

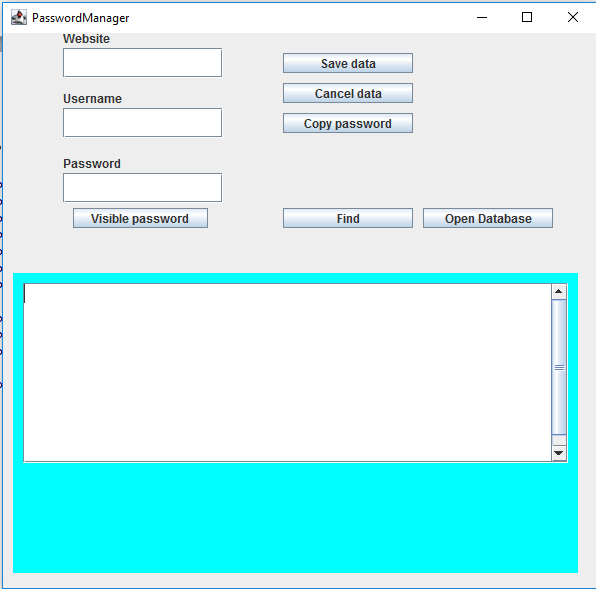
Find password

Open Database

**Код програми на мові Java:**

**import** javax.swing.\*;  
**import** java.awt.\*;  
**import** java.awt.event.ActionEvent;  
**import** java.awt.event.ActionListener;  
**import** java.io.\*;  
  
**public class** FramePassword\_22\_10 {  
 **public static void** main(String[] args) {  
 JFrame frame = **new** JFrame(**"PasswordManager"**);  
 JPanel panel = **new** JPanel();  
  
 frame.setSize(600, 550);  
 panel.setBounds(10, 240, 565, 300);  
 panel.setBackground(Color.***cyan***);  
  
 JTextField webSiteField = **new** JTextField();  
 Font font = **new** Font(**"Courier New"**, Font.***BOLD***, 16); *// Courier New - моноширинный шрифт* webSiteField.setFont(font);  
 JTextField userNameField = **new** JTextField();  
 userNameField.setFont(font);  
 JPasswordField passwordField = **new** JPasswordField();  
 passwordField.setFont(font);  
  
 webSiteField.setBounds(60, 15, 160, 30);  
 userNameField.setBounds(60, 75, 160, 30);  
 passwordField.setBounds(60, 140, 160, 30);  
  
 JLabel label1 = **new** JLabel(**"Website"**);  
 JLabel label2 = **new** JLabel(**"Username"**);  
 JLabel label3 = **new** JLabel(**"Password"**);  
 label1.setBounds(60, 0, 80, 10);  
 label2.setBounds(60, 60, 80, 10);  
 label3.setBounds(60, 125, 80, 10);  
  
 JButton saveButton = **new** JButton(**"Save data"**);  
 JButton cancelButton = **new** JButton(**"Cancel data"**);  
 JButton visiblePasswordButton = **new** JButton(**"Visible password"**);  
 JButton openDatabaseButton = **new** JButton(**"Open Database"**);  
 JButton copyPasswordButton = **new** JButton(**"Copy password"**);  
 JButton findButton = **new** JButton(**"Find"**);  
 findButton.setBounds(280, 175, 130, 20);  
 frame.add(findButton);  
  
 saveButton.setBounds(280, 20, 130, 20);  
 cancelButton.setBounds(280, 50, 130, 20);  
 visiblePasswordButton.setBounds(70, 175, 135, 20);  
 openDatabaseButton.setBounds(420, 175, 130, 20);  
 copyPasswordButton.setBounds(280, 80, 130, 20);  
  
 frame.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);  
 frame.add(webSiteField);  
 frame.add(userNameField);  
 frame.add(passwordField);  
 frame.add(panel);  
 frame.add(label1);  
 frame.add(label2);  
 frame.add(label3);  
 frame.add(saveButton);  
 frame.add(cancelButton);  
 frame.add(visiblePasswordButton);  
 frame.add(openDatabaseButton);  
 frame.add(copyPasswordButton);  
 frame.setLayout(**null**);  
 panel.setLayout(**null**);  
 frame.setVisible(**true**);  
  
 PasswordManager passwordManager = **new** PasswordManager(panel, frame);  
  
 saveButton.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
 passwordManager.saveDataTextFields(userNameField,passwordField, webSiteField);  
 }  
 });  
  
 cancelButton.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
 passwordManager.cancelDataTextFields(userNameField,passwordField, webSiteField );  
 }  
 });  
 visiblePasswordButton.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
  
 passwordManager.showVisiblePassword(passwordField);  
 }  
 });  
  
 openDatabaseButton.addActionListener(**new** ActionListener() {  
 **private int i** = 0; *// Локальная переменная для счетчика* @Override  
 **public void** actionPerformed(ActionEvent e) {  
 **if** (**i** % 2 == 0) {  
 passwordManager.openDatabase(panel);  
 } **else** {  
 passwordManager.openClearDatabase(panel);  
 }  
 **i**++; *// Увеличиваем i после каждого нажатия кнопки* }  
 });  
 copyPasswordButton.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
 String password = **new** String(passwordField.getPassword());  
  
 JTextField tempField = **new** JTextField();  
 tempField.setText(password);  
 tempField.selectAll();  
 tempField.copy();  
 }  
 });  
 findButton.addActionListener(**new** ActionListener() {  
 @Override  
 **public void** actionPerformed(ActionEvent e) {  
 String targetWebsite = webSiteField.getText();  
  
  
 *// Вызов метода для поиска пароля и вывода его в соответствующее поле* passwordManager.findAndDisplayUsername(targetWebsite, userNameField, passwordField);  
 }  
 });  
  
 }  
}  
**class** PasswordManager **extends** JFrame {  
  
 **private** JFileChooser **fileChooser**;  
 **private** JTextArea **textArea**;  
 **private** JFrame **frame**;  
 **private boolean passwordVisible** = **false**;  
  
 **public** PasswordManager(JPanel panel, JFrame frame) {  
 **textArea** = **new** JTextArea(10, 40);  
 Font font = **new** Font(**"Courier New"**, Font.***BOLD***, 16); *// Courier New - моноширинный шрифт* **textArea**.setFont(font);  
 JScrollPane scrollPane = **new** JScrollPane(**textArea**);  
 scrollPane.setBounds(10, 10, 545, 180);  
 panel.add(scrollPane);  
 **fileChooser** = **new** JFileChooser();  
 **this**.**frame** = frame;  
 }  
  
 **public void** saveDataTextFields(JTextField userName, JPasswordField passwordField, JTextField webSite) {  
 String user = userName.getText();  
 String password = **new** String(passwordField.getPassword());  
 String website = webSite.getText();  
 **int** returnValue = **fileChooser**.showSaveDialog(**null**);  
 File selectedFile = **null**;  
 **if** (returnValue == JFileChooser.***APPROVE\_OPTION***) {  
 selectedFile = **fileChooser**.getSelectedFile();  
 }  
  
 **if** (!user.isEmpty() && !password.isEmpty() && !website.isEmpty()) {  
 **try** (BufferedWriter writer = **new** BufferedWriter(**new** FileWriter(selectedFile, **true**))) {  
 String formattedText = String.*format*(**"\nWebSite: %-20s | UserName: %-20s | Password: %s%n"**, website, user, password);  
 writer.write(formattedText);  
 writer.close();  
 JOptionPane.*showMessageDialog*(**null**, **"Пароль сохранен."**);  
 clearFields(userName, passwordField, webSite);  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 JOptionPane.*showMessageDialog*(**null**, **"Ошибка при сохранении пароля."**);  
 }  
 } **else** {  
 JOptionPane.*showMessageDialog*(**null**, **"Пожалуйста, заполните все поля."**);  
 }  
 }  
  
  
 **public void** cancelDataTextFields(JTextField userName, JPasswordField passwordField, JTextField webSite) {  
 clearFields(userName, passwordField, webSite);  
 }  
  
 **public void** showVisiblePassword(JPasswordField passwordField) {  
 **char**[] password = passwordField.getPassword();  
 String passwordText = **new** String(password);  
  
 **if** (**passwordVisible**) {  
 passwordField.setEchoChar(**'\u2022'**); *// Пароль приховується (символ для приховування паролю)* } **else** {  
 passwordField.setEchoChar((**char**) 0); *// Пароль робиться видимим* }  
  
 passwordField.setText(passwordText);  
 **passwordVisible** = !**passwordVisible**; *// Змінюємо стан видимості паролю* }  
  
 **public void** clearFields(JTextField userName, JPasswordField passwordField, JTextField webSite) {  
 userName.setText(**""**);  
 passwordField.setText(**""**);  
 webSite.setText(**""**);  
 }  
  
 **public void** openDatabase(JPanel panel) {  
 **int** returnValue = **fileChooser**.showOpenDialog(**null**);  
  
 **if** (returnValue == JFileChooser.***APPROVE\_OPTION***) {  
 File selectedFile = **fileChooser**.getSelectedFile();  
  
 String currentFileName = selectedFile.getAbsolutePath();  
 **try** (BufferedReader reader = **new** BufferedReader(**new** FileReader(selectedFile))) {  
 String line;  
 StringBuilder content = **new** StringBuilder();  
 **while** ((line = reader.readLine()) != **null**) {  
 content.append(line).append(**"\n"**);  
 }  
  
 **textArea**.setText(content.toString());  
 **frame**.setTitle(**"Base data - "** + currentFileName);  
 } **catch** (IOException ex) {  
 ex.printStackTrace();  
 }  
 }  
 }  
  
 **public void** openClearDatabase(JPanel panel) {  
 **textArea**.setText(**""**); *// Очищаем вміст textArea* **frame**.setTitle(**"Password Manager"** );  
 }  
  
 **public void** findAndDisplayUsername(String targetWebsite, JTextField userField, JTextField passwordField) {  
 String text = **textArea**.getText();  
 String[] lines = text.split(**"\n"**);  
  
 **int** totalWords = 0;  
 **for** (String line : lines) {  
 String[] wordsInLine = line.split(**" "**);  
 totalWords += wordsInLine.**length**;  
 }  
  
 String[] allWords = **new** String[totalWords];  
 **int** index = 0;  
  
 **for** (String line : lines) {  
 String[] wordsInLine = line.split(**" "**);  
 **for** (String word : wordsInLine) {  
 allWords[index] = word;  
 index++;  
 }  
 }  
  
 String username = **""**;  
 String password =**""**;  
 **for** (**int** i = 0; i < allWords.**length**; i++) {  
 String currentWord = allWords[i];  
  
 **if** (currentWord.equalsIgnoreCase(**"WebSite:"**)) {  
 *// Найдено слово "WebSite:", следующее за ним - адрес сайта* String website = allWords[i + 1];  
  
 **if** (website.equalsIgnoreCase(targetWebsite)) {  
 *// Найдено совпадение между введенным именем веб-сайта и текстом в текстовой области  
 // Теперь ищем имя пользователя* **for** (**int** j = i; j < allWords.**length**; j++) {  
 **if** (allWords[j].equalsIgnoreCase(**"UserName:"**)) {  
 *// Найдено слово "UserName:", следующее за ним - имя пользователя* username = allWords[j + 1];  
 }  
 **if** (allWords[j].equalsIgnoreCase(**"Password:"**)){  
 password =allWords[j+1];  
 **break**; *// Мы нашли имя пользователя, поэтому нет нужды продолжать поиск* }  
 }  
 }  
 }  
 }  
  
 userField.setText(username);  
 passwordField.setText(password);  
  
 }  
}

**Результат виконання програми:**



**Висновок:** в цій практичній роботі розроблено програму-менеджер паролів з графічним інтерфейсом користувача.

Основні елементи інтерфейсу включають поля для введення веб-сайту, імені користувача та паролю, кнопки для збереження, скасування, відображення/приховування паролю, відкриття та закриття бази даних паролів, копіювання паролю та пошуку. Реалізовано можливість введення та збереження даних для кожного веб-сайту (веб-сайт, ім'я користувача, пароль). Забезпечено можливість скасування введених даних та очищення полів вводу.

Реалізовано функціонал відображення або приховування паролю для перевірки його правильності. Додана кнопка "Copy password" для копіювання введеного паролю у буфер обміну.

Практична робота успішно виконана, програма-менеджер паролів відповідає вимогам технічного завдання та володіє необхідним функціоналом для безпечного зберігання та управління паролями. Інтерфейс користувача забезпечує зручність використання, а реалізовані функції дозволяють ефективно взаємодіяти з програмою.