**Paraschivoiu Alexandru Daniel**

**Grupa 222 – Proiect Java – High/Low**

**Structura Aplicatiei:**

Aceasta aplicatie este un joc High Low cu interfata grafica (GUI) construit in Java. Clasa Login este utilizata pentru a afisa un formular de login pentru utilizator.

Formularul contine campuri pentru nume si parola si un buton de login. Daca numele si parola sunt corecte, utilizatorul este logat si fereastra de login se inchide. In caz contrar, utilizatorul primeste un mesaj de eroare.   
Clasa Login are o metoda displayForm() care afiseaza formularul de login si o metoda isLoggedIn() care returneaza true daca utilizatorul este logat si false in caz contrar.

Clasa HighLowGUI este utilizata pentru a afisa si gestiona interfata grafica a jocului High/Low. Jocul consta in a ghici daca urmatoarea carte va fi mai mare sau mai mica decat cartea curenta.

Jocul poate fi jucat de 2 jucători, iar scorul fiecarui jucator este afisat pe ecran. Inainte de a incepe jocul, utilizatorul trebuie sa se logheze folosind formularul de login din clasa Login.

Clasa HighLowGUI are doua clase interne HighButtonListener si LowButtonListener care gestioneaza evenimentele declansate de butoanele high si low.

import java.awt.Component;  
import java.awt.GridLayout;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.ArrayList;  
import java.util.Collections;  
import javax.swing.\*;  
  
  
  
*/\*\*  
  
 Clasa Login este utilizata pentru a afisa un formular de login pentru utilizator.  
  
 Formularul contine campuri pentru nume si parola si un buton de login.  
  
 Daca numele si parola sunt corecte, utilizatorul este logat si fereastra de login se inchide.  
  
 In caz contrar, utilizatorul primeste un mesaj de eroare.  
 \*/*class Login {  
 private JFrame loginFrame; // fereastra de login  
 private JTextField numeField; // campul pentru nume  
 private JPasswordField parolaField; // campul pentru parola  
 private JButton loginButton; // butonul de login  
 private boolean isLoggedIn = false; // variabila care retine daca utilizatorul este logat  
  
 */\*\*  
  
 Metoda displayForm afiseaza formularul de login.  
 \*/* public void displayForm() {  
 loginFrame = new JFrame("Login");  
 loginFrame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
 JPanel loginPanel = new JPanel();  
 loginPanel.setLayout(new GridLayout(3, 2));  
 JLabel numeLabel = new JLabel("Nume: ");  
 JLabel parolaLabel = new JLabel("Parola: ");  
 numeField = new JTextField();  
 parolaField = new JPasswordField();  
 loginButton = new JButton("Login");  
 loginButton.addActionListener(new LoginButtonListener());  
 loginPanel.add(numeLabel);  
 loginPanel.add(numeField);  
 loginPanel.add(parolaLabel);  
 loginPanel.add(parolaField);  
 loginPanel.add(loginButton);  
 loginFrame.add(loginPanel);  
 loginFrame.pack();  
 loginFrame.setLocationRelativeTo(null); // dam center la login frame  
 loginFrame.setVisible(true);  
 }  
 */\*\*  
  
 Metoda isLoggedIn returneaza true daca utilizatorul este logat si false in caz contrar.* ***@return*** *true daca utilizatorul este logat, false altfel.  
 \*/* public boolean isLoggedIn() {  
 return isLoggedIn;  
 }  
 */\*\*  
  
 Clasa LoginButtonListener este utilizata pentru a procesa evenimentul de apasare a butonului de login.  
  
 Daca numele si parola introdusa sunt corecte, utilizatorul este logat si fereastra de login se inchide.  
  
 In caz contrar, utilizatorul primeste un mesaj de eroare.  
 \*/* private class LoginButtonListener implements ActionListener {  
 private LoginButtonListener() {  
 }  
  
 public void actionPerformed(ActionEvent e) {  
 String nume = Login.this.numeField.getText();  
 String parola = new String(Login.this.parolaField.getPassword());  
 if (nume.equals("paraschivoiu") && parola.equals("alexandru")) {  
 Login.this.isLoggedIn = true;  
 Login.this.loginFrame.dispose();  
 } else {  
 JOptionPane.*showMessageDialog*((Component)null, "Nume sau parola gresita");  
 }  
 }  
 }  
}  
  
*/\*\*  
  
 Clasa HighLowGUI este utilizata pentru a afisa si gestiona interfata grafica a jocului High/Low.  
  
 Jocul consta in a ghici daca urmatoarea carte va fi mai mare sau mai mica decat cartea curenta.  
  
 Jocul poate fi jucat de 2 jucatori, iar scorul fiecarui jucator este afisat pe ecran.  
  
 Inainte de a incepe jocul, utilizatorul trebuie sa se logheze folosind formularul de login.  
 \*/*public class HighLowGUI {  
 private ArrayList<Integer> pachet; // pachetul de carti  
 private int carteaCurenta; // cartea curenta  
 private int carteaUrmatoare; // cartea urmatoare  
 private JLabel carteaCurentaLabel; // eticheta pentru cartea curenta  
 private JLabel carteaUrmatoareLabel; // eticheta pentru cartea urmatoare  
 private JButton highButton; // butonul pentru a alege "high"  
 private JButton lowButton; // butonul pentru a alege "low"  
 private int jucator1Scor = 0; // scorul jucatorului 1  
 private int jucator2Scor = 0; // scorul jucatorului 2  
 private JLabel jucator1ScorLabel; // eticheta pentru scorul jucatorului 1  
 private JLabel jucator2ScorLabel; // eticheta pentru scorul jucatorului 2  
 private String jucatorCurent; // numele jucatorului curent  
 private ButtonGroup jucatorGroup; // grupul de butoane pentru a selecta jucatorul curent  
 private JRadioButton jucator1Button; // butonul pentru jucatorul 1  
 private JRadioButton jucator2Button; // butonul pentru jucatorul 2  
  
 */\*\*  
  
 Constructorul pentru clasa HighLowGUI afiseaza formularul de login si initializeaza interfata grafica pentru joc.  
 \*/* public HighLowGUI() {  
 jucator1ScorLabel = new JLabel("Jucator 1: " + jucator1Scor);  
 jucator2ScorLabel = new JLabel("Jucator 2: " + jucator2Scor);  
 jucatorGroup = new ButtonGroup();  
 jucator1Button = new JRadioButton("Jucator 1");  
 jucator2Button = new JRadioButton("Jucator 2");  
 Login login = new Login();  
 login.displayForm();  
 while(!login.isLoggedIn()) {  
 pachet = new ArrayList<>();  
 for(int i = 2; i <= 14; ++i) {  
 for(int j = 0; j < 4; ++j) {  
 pachet.add(i);  
 }  
 }  
  
  
 Collections.*shuffle*(pachet);  
 carteaCurenta = (Integer) pachet.get(0);  
 pachet.remove(0);  
 carteaUrmatoare = (Integer) pachet.get(0);  
 pachet.remove(0);  
 carteaCurentaLabel = new JLabel("CarteaCurenta: " + carteaCurenta);  
 carteaUrmatoareLabel = new JLabel();  
 highButton = new JButton("High");  
 lowButton = new JButton("Low");  
 highButton.addActionListener(new HighButtonListener());  
 lowButton.addActionListener(new LowButtonListener());  
 jucatorGroup.add(jucator1Button);  
 jucatorGroup.add(jucator2Button);  
 jucator1Button.addActionListener(new PlayerSelectListener());  
 jucator2Button.addActionListener(new PlayerSelectListener());  
  
 }  
  
 JPanel jocPanel = new JPanel();  
 jocPanel.setLayout(new GridLayout(4, 2));  
 jocPanel.add(carteaCurentaLabel);  
 jocPanel.add(carteaUrmatoareLabel);  
 jocPanel.add(highButton);  
 jocPanel.add(lowButton);  
 jocPanel.add(jucator1Button);  
 jocPanel.add(jucator2Button);  
 jocPanel.add(jucator1ScorLabel);  
 jocPanel.add(jucator2ScorLabel);  
 JFrame jocFrame = new JFrame("Joc High/Low ");  
 jocFrame.setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
 jocFrame.add(jocPanel);  
 jocFrame.pack();  
 jocFrame.setLocationRelativeTo(null); // center la joc frame  
 jocFrame.setVisible(true);  
 }  
  
 public static void main(String[] args) {  
 new HighLowGUI();  
 }  
  
 */\*\*  
 \* Clasa HighButtonListener este utilizata pentru a procesa evenimentul de apasare a butonului "High".  
 \* Daca alegerea jucatorului este corecta, scorul acestuia este actualizat si se trece la urmatoarea carte.  
 \* In caz contrar, jucatorul pierde si se trece la urmatorul jucator.  
 \*/* private class HighButtonListener implements ActionListener {  
 private HighButtonListener() {  
 }  
  
 public void actionPerformed(ActionEvent e) {  
 if (HighLowGUI.this.carteaUrmatoare > HighLowGUI.this.carteaCurenta) {  
 if (HighLowGUI.this.jucatorCurent.equals("Jucator 1")) {  
 ++HighLowGUI.this.jucator1Scor;  
 HighLowGUI.this.jucator1ScorLabel.setText("Jucator 1: " + HighLowGUI.this.jucator1Scor);  
 } else {  
 ++HighLowGUI.this.jucator2Scor;  
 HighLowGUI.this.jucator2ScorLabel.setText("Jucator 2: " + HighLowGUI.this.jucator2Scor);  
 }  
  
 JOptionPane.*showMessageDialog*((Component)null, "Corect! Cartea ghicita este " + HighLowGUI.this.carteaUrmatoare);  
 } else if (HighLowGUI.this.carteaUrmatoare == HighLowGUI.this.carteaCurenta) {  
 JOptionPane.*showMessageDialog*((Component)null, "Egalitate! Cartea ghicita este " + HighLowGUI.this.carteaUrmatoare);  
 }  
  
 else {  
 JOptionPane.*showMessageDialog*((Component)null, "Gresit! Cartea ghicita este " + HighLowGUI.this.carteaUrmatoare);  
 }  
  
  
  
 if (HighLowGUI.this.jucator1Scor == 5) {  
 JOptionPane.*showMessageDialog*((Component)null, "Jucatorul 1 a castigat!");  
 } else if (HighLowGUI.this.jucator2Scor == 5) {  
 JOptionPane.*showMessageDialog*((Component)null, "Jucatorul 2 a castigat!");  
 }  
  
 Collections.*shuffle*(HighLowGUI.this.pachet);  
 HighLowGUI.this.carteaCurenta = (Integer)HighLowGUI.this.pachet.get(0);  
 HighLowGUI.this.pachet.remove(0);  
 HighLowGUI.this.carteaUrmatoare = (Integer)HighLowGUI.this.pachet.get(0);  
 HighLowGUI.this.pachet.remove(0);  
 HighLowGUI.this.carteaCurentaLabel.setText("CarteaCurenta: " + HighLowGUI.this.carteaCurenta);  
 }  
 }  
 */\*\*  
 \* Clasa LowButtonListener este utilizata pentru a procesa evenimentul de apasare a butonului "Low".  
 \* Daca alegerea jucatorului este corecta, scorul acestuia este actualizat si se trece la urmatoarea carte.  
 \* In caz contrar, jucatorul pierde si se trece la urmatorul jucator.  
 \*/* private class LowButtonListener implements ActionListener {  
 private LowButtonListener() {  
 }  
  
 public void actionPerformed(ActionEvent e) {  
 if (HighLowGUI.this.carteaUrmatoare < HighLowGUI.this.carteaCurenta) {  
 if (HighLowGUI.this.jucatorCurent.equals("Jucator 1")) {  
 ++HighLowGUI.this.jucator1Scor;  
 HighLowGUI.this.jucator1ScorLabel.setText("Jucator 1: " + HighLowGUI.this.jucator1Scor);  
 } else {  
 ++HighLowGUI.this.jucator2Scor;  
 HighLowGUI.this.jucator2ScorLabel.setText("Jucator 2: " + HighLowGUI.this.jucator2Scor);  
 }  
  
 JOptionPane.*showMessageDialog*((Component)null, "Corect! Cartea ghicita este " + HighLowGUI.this.carteaUrmatoare);  
 } else if (HighLowGUI.this.carteaUrmatoare == HighLowGUI.this.carteaCurenta) {  
 JOptionPane.*showMessageDialog*((Component)null, "Egalitate! Cartea ghicita este " + HighLowGUI.this.carteaUrmatoare);  
 }  
  
 else {  
 JOptionPane.*showMessageDialog*((Component)null, "Gresit! Cartea ghicita este " + HighLowGUI.this.carteaUrmatoare);  
 }  
  
  
 if (HighLowGUI.this.jucator1Scor == 5) {  
 JOptionPane.*showMessageDialog*((Component)null, "Jucatorul 1 a castigat!");  
 } else if (HighLowGUI.this.jucator2Scor == 5) {  
 JOptionPane.*showMessageDialog*((Component)null, "Jucatorul 2 a castigat!");  
 }  
  
 Collections.*shuffle*(HighLowGUI.this.pachet);  
 HighLowGUI.this.carteaCurenta = (Integer)HighLowGUI.this.pachet.get(0);  
 HighLowGUI.this.pachet.remove(0);  
 HighLowGUI.this.carteaUrmatoare = (Integer)HighLowGUI.this.pachet.get(0);  
 HighLowGUI.this.pachet.remove(0);  
 HighLowGUI.this.carteaCurentaLabel.setText("CarteaCurenta: " + HighLowGUI.this.carteaCurenta);  
 }  
 }  
 private class PlayerSelectListener implements ActionListener {  
 private PlayerSelectListener() {  
 }  
 public void actionPerformed(ActionEvent e) {  
 HighLowGUI.this.jucatorCurent = e.getActionCommand();  
 }  
 }  
}