МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

кафедра програмних засобів

Звіт

з лабораторної роботи № 4

з дисципліни «Мультипарадигмальне програмування»

на тему:

«Компонентно-орієнтоване програмування»

Виконав

Ст. гр. КНТ-137 Козлов В.В.

Прийняв

Качан О.І

1. Мета роботи

Ознайомитись з принципами компонентно-орієнтованого програмування. Навчитися використовувати компонентні технології в процесі розроблення програмних систем.

1. Завдання на лабораторну роботу
   1. Варіант 9

Ознайомитись з теоретичними відомостями, необхідними для виконання роботи.

Розробити програмне забезпечення у відповідності з індивідуальним завданням, узгодженим з викладачем. Усі функції системи повинні бути реалізовані у вигляді необхідної кількості програмних компонентів .NET мовою програмування C#, для чого потрібно ще до реалізації забезпечити відповідний процес проектування з обґрунтуванням отриманих результатів. Клієнтське застосування повинно використовувати функції з компонентів через програмні інтерфейси. Програмне забезпечення обов’язково має відповідати вимогам безпеки даних та програмного коду.

* + 1. Завдання а
  1. Текст завдання

Розробити систему, яка реалізує футбольний тоталізатор. Адміністратор визначає футбольні матчі, дати їх проведення та варіанти виграшів для можливих результатів (результат матчу, кількість забитих м’ячів). Користувачі роблять ставки, визначаючи результат та суму грошей. Система визначає результати матчів випадковим чином після настання відповідної дати. За отриманими результатами визначається виграш кожного учасника та надсилається йому відповідне повідомлення. Система повинна надавати можливість реєстрації та авторизації. Для кожного гравця накопичується певна сума за результатами його участі.

* 1. Вміст файлу SystemManagment.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Text.RegularExpressions;

namespace BetSystem

{

public interface IUserInterface

{

bool sign\_in(string Login, string Password);

bool sign\_up(string Login, string Password);

string get\_money();

string get\_wins();

string get\_login();

string[] get\_match\_coefficient(string MatchNumber);

bool new\_bet(string MatchNumber, string BetInfo, string MoneyToBet);

List<string[]> get\_matches();

List<string[]> get\_history();

bool plus\_one\_coin();

}

public interface IAdminInterface

{

bool sign\_in\_admin(string Login, string Password);

bool add\_match(string Team1, string Team2, string Date, string Coefficients);

List<string[]> get\_matches();

bool update\_match(string MatchNumber, decimal FirstTeamGoals, decimal SecondTeamGoals);

}

public class SystemManagment : IUserInterface, IAdminInterface

{

string FileUsers = @"user.txt";

string FileMatches = @"match.txt";

string FileBet = @"bet.txt";

string FileUserBets = @"userbets.txt";

public SystemManagment()

{

using (StreamWriter FileWrite = new StreamWriter(FileUsers, true)) ;

using (StreamWriter FileWrite = new StreamWriter(FileMatches, true)) ;

using (StreamWriter FileWrite = new StreamWriter(FileBet, true)) ;

using (StreamWriter FileWrite = new StreamWriter(FileUserBets, true)) ;

}

string Name = "";

public bool sign\_in(string Login, string Password)

{

Name = Login;

try

{

using (StreamReader FileRead = new StreamReader(FileUsers))

{

string[] Lines = FileRead.ReadToEnd().Split(new char[] { '\r', '\n' }, StringSplitOptions.RemoveEmptyEntries);

foreach (string Line in Lines)

{

string LoginFromFile = Line.Split(' ')[0];

string PasswordFromFile = Line.Split(' ')[1];

if (LoginFromFile == Login && Password == PasswordFromFile)

{

return true;

}

}

}

}

catch (Exception)

{

using (StreamWriter FileWrite = new StreamWriter(FileUsers, true)) ;

}

return false;

}

private void write\_str\_to\_file(string FileName, string Data)

{

using (StreamWriter FileWrite = new StreamWriter(FileName, true))

{

FileWrite.WriteLine(Data);

}

}

public bool sign\_up(string Login, string Password)

{

try

{

if(Login == "admin")

{

return false;

}

using (StreamReader FileRead = new StreamReader(FileUsers))

{

string[] Lines = FileRead.ReadToEnd().Split(new char[] { '\r', '\n' }, StringSplitOptions.RemoveEmptyEntries);

foreach (string Line in Lines)

{

string LoginFromFile = Line.Split(' ')[0];

if (LoginFromFile == Login)

{

return false;

}

}

}

}

catch(Exception)

{

using (StreamWriter FileWrite = new StreamWriter(FileUsers, true)) ;

}

write\_str\_to\_file(FileUsers, Login + ' ' + Password + ' ' + "1000");

return true;

}

public string get\_money()

{

try

{

using (StreamReader FileRead = new StreamReader(FileUsers))

{

string[] Lines = FileRead.ReadToEnd().Split(new char[] { '\r', '\n' }, StringSplitOptions.RemoveEmptyEntries);

foreach (string Line in Lines)

{

string LoginFromFile = Line.Split(' ')[0];

if (LoginFromFile == Name)

{

return Line.Split(' ')[2];

}

}

}

return "0";

}

catch (Exception)

{

return "0";

};

}

public string get\_wins()

{

try

{

int Wins = 0;

using (StreamReader FileRead = new StreamReader(FileUserBets))

{

string[] Lines = FileRead.ReadToEnd().Split(new char[] { '\r', '\n' }, StringSplitOptions.RemoveEmptyEntries);

foreach (string Line in Lines)

{

string LoginFromFile = Line.Split(' ')[1];

if (LoginFromFile == Name)

{

string Result = Line.Split(' ')[4];

if (Result == "WIN")

{

Wins++;

}

}

}

}

return Wins.ToString();

}

catch (Exception)

{

return "111";

};

}

public string get\_login()

{

return Name;

}

public string[] get\_match\_coefficient(string MatchNumber)

{

try

{

using (StreamReader FileRead = new StreamReader(FileBet))

{

string[] Lines = FileRead.ReadToEnd().Split(new char[] { '\r', '\n' }, StringSplitOptions.RemoveEmptyEntries);

foreach (string Line in Lines)

{

string Number = Line.Split(' ')[0];

string[] Bet = Line.Split(' ');

if (Number == MatchNumber)

{

string[] coefficient = new string[8];

for(int i = 0; i < 8; i++)

{

coefficient[i] = Bet[i + 1];

}

return coefficient;

}

}

}

}

catch (Exception)

{

using (StreamWriter FileWrite = new StreamWriter(FileUsers, true)) ;

}

return null;

}

public bool sign\_in\_admin(string Login, string Password)

{

Name = Login;

if (Login == "admin" && Password == "admin") return true;

else return false;

}

public List<string[]> get\_matches()

{

try

{

List<string[]> Matches = new List<string[]>();

using (StreamReader FileRead = new StreamReader(FileMatches))

{

string[] Lines = FileRead.ReadToEnd().Split(new char[] { '\r', '\n' }, StringSplitOptions.RemoveEmptyEntries);

foreach (string Line in Lines)

{

if(Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[4] != "+")

{

string[] Match = new string[3];

Match[0] = Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[0];

Match[1] = Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[1];

Match[2] = Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[2] + " - " + Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[3];

Matches.Add(Match);

}

}

}

return Matches;

}

catch (Exception)

{

return null;

}

}

private string get\_resultSum(string MatchNumber, string BetInfo, string Sum)

{

string coefficient = find\_coefficient(MatchNumber, BetInfo);

string ResultSum = Convert.ToInt32((Convert.ToDecimal(Sum) \* Convert.ToDecimal(coefficient))).ToString();

return ResultSum;

}

public List<string[]> get\_history()

{

try

{

List<string[]> History = new List<string[]>();

using (StreamReader FileRead = new StreamReader(FileUserBets))

{

string[] Lines = FileRead.ReadToEnd().Split(new char[] { '\r', '\n' }, StringSplitOptions.RemoveEmptyEntries);

foreach (string Line in Lines)

{

if(Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[1] == Name)

{

string coefficient = find\_coefficient(Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[0], Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[2]);

string ResultSum = (Convert.ToDecimal(Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[3]) \* Convert.ToDecimal(coefficient)).ToString("0.00");

string[] Bet = new string[6];

Bet[0] = Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[0];

Bet[1] = Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[2];

Bet[2] = find\_coefficient(Bet[0], Bet[1]);

Bet[3] = Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[3];

Bet[4] = get\_resultSum(Bet[0], Bet[1], Bet[3]);

Bet[5] = Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[4];

History.Add(Bet);

}

}

}

History.Reverse();

return History;

}

catch (Exception)

{

using (StreamWriter FileWriter = new StreamWriter(FileUserBets)) ;

List<string[]> History = new List<string[]>();

return History;

}

}

private string find\_coefficient(string MatchNumber, string BetInfo)

{

try

{

using (StreamReader FileRead = new StreamReader(FileBet))

{

string[] Lines = FileRead.ReadToEnd().Split(new char[] { '\r', '\n' }, StringSplitOptions.RemoveEmptyEntries);

foreach (string Line in Lines)

{

if(Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[0] == MatchNumber)

{

if (BetInfo == "W1") return Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[1];

else if (BetInfo == "D") return Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[2];

else if (BetInfo == "W2") return Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[3];

else if (BetInfo == "T0") return Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[4];

else if (BetInfo == "T1") return Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[5];

else if (BetInfo == "T2") return Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[6];

else if (BetInfo == "T3") return Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[7];

else if (BetInfo == "T4Plus") return Line.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries)[8];

}

}

}

return null;

}

catch (Exception)

{

return null;

}

}

public bool new\_bet(string MatchNumber, string BetInfo, string MoneyToBet)

{

try

{

bool IsBetConfirmed = false;

using (StreamReader FileRead = new StreamReader(FileUsers))

{

string[] Lines = FileRead.ReadToEnd().Split(new char[] { '\r', '\n' }, StringSplitOptions.RemoveEmptyEntries);

foreach (string Line in Lines)

{

string LoginFromFile = Line.Split(' ')[0];

if (LoginFromFile == Name)

{

if (Convert.ToDecimal(MoneyToBet) > 0 && Convert.ToDecimal(Line.Split(' ')[2]) >= Convert.ToDecimal(MoneyToBet))

{

IsBetConfirmed = true;

write\_str\_to\_file(FileUserBets, MatchNumber + ' ' + Name + ' ' + BetInfo + ' ' + MoneyToBet + " WAIT");

break;

}

else

{

return false;

}

}

}

}

if (IsBetConfirmed)

{

return update\_user\_money(Name, "-" + MoneyToBet);

}

else

{

return false;

}

}

catch (Exception)

{

return false;

}

}

public bool plus\_one\_coin()

{

return update\_user\_money(Name, "1");

}

private bool update\_user\_money(string UserName, string AddMoney)

{

try

{

string[] Users;

using (StreamReader FileRead = new StreamReader(FileUsers))

{

Users = FileRead.ReadToEnd().Split(new char[] {'\r', '\n'}, StringSplitOptions.RemoveEmptyEntries);

}

for (int i = 0; i < Users.Count(); i++)

{

if(Users[i].Split(' ')[0] == UserName)

{

string NewSum = (Convert.ToDecimal(Users[i].Split(' ')[2]) + Convert.ToDecimal(AddMoney)).ToString();

Users[i] = Users[i].Split(' ')[0] + ' ' + Users[i].Split(' ')[1] + ' ' + NewSum;

}

}

using (StreamWriter FileWrite = new StreamWriter(FileUsers, false))

{

foreach(string User in Users)

{

FileWrite.WriteLine(User);

}

}

return true;

}

catch(Exception)

{

return false;

}

}

public bool add\_match(string Team1, string Team2, string Date, string Coefficients)

{

string MatchNumber;

try

{

using (StreamReader FileRead = new StreamReader(FileMatches))

{

MatchNumber = (FileRead.ReadToEnd().Split(new char[] { '\r', '\n' }, StringSplitOptions.RemoveEmptyEntries).Count() + 1).ToString();

}

write\_str\_to\_file(FileBet, MatchNumber + ' ' + Coefficients);

write\_str\_to\_file(FileMatches, MatchNumber + ' ' + Date + ' ' + Team1 + ' ' + Team2 + ' ' + "- 0 0");

}

catch(Exception)

{

return false;

}

return true;

}

public bool update\_match(string MatchNumber, decimal FirstTeamGoals, decimal SecondTeamGoals)

{

try

{

string[] lines;

using (StreamReader FileRead = new StreamReader(FileMatches))

{

lines = FileRead.ReadToEnd().Split(new char[] { '\r', '\n' }, StringSplitOptions.RemoveEmptyEntries);

}

for (int i = 0; i < lines.Count(); i++)

{

string[] Match = lines[i].Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries);

if (Match[0] == MatchNumber && Match[4] == "-")

{

Match[4] = "+";

Match[5] = FirstTeamGoals.ToString();

Match[6] = SecondTeamGoals.ToString();

lines[i] = string.Join(" ", Match);

break;

}

}

using (StreamWriter FileWriter = new StreamWriter(FileMatches, false))

{

foreach(string Match in lines)

{

FileWriter.WriteLine(Match);

}

}

using (StreamReader FileRead = new StreamReader(FileUserBets))

{

lines = FileRead.ReadToEnd().Split(new char[] { '\r', '\n' }, StringSplitOptions.RemoveEmptyEntries);

}

for(int i = 0; i < lines.Count(); i++)

{

string[] Bet = lines[i].Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries);

if(Bet[0] == MatchNumber)

{

bool isWin = false;

if(Bet[2] == "W1" && FirstTeamGoals > SecondTeamGoals)

{

isWin = true;

}

if (Bet[2] == "D" && FirstTeamGoals == SecondTeamGoals)

{

isWin = true;

}

if (Bet[2] == "W2" && FirstTeamGoals < SecondTeamGoals)

{

isWin = true;

}

if (Bet[2] == "T0" && FirstTeamGoals + SecondTeamGoals == 0)

{

isWin = true;

}

if (Bet[2] == "T1" && FirstTeamGoals + SecondTeamGoals == 1)

{

isWin = true;

}

if (Bet[2] == "T2" && FirstTeamGoals + SecondTeamGoals == 2)

{

isWin = true;

}

if (Bet[2] == "T3" && FirstTeamGoals + SecondTeamGoals == 3)

{

isWin = true;

}

if (Bet[2] == "T4Plus" && FirstTeamGoals + SecondTeamGoals > 3)

{

isWin = true;

}

if (isWin)

{

string coefficient = find\_coefficient(MatchNumber, Bet[2]);

string ResultSum = (Convert.ToDecimal(Bet[3]) \* Convert.ToDecimal(coefficient)).ToString("0.00");

update\_user\_money(Bet[1], get\_resultSum(MatchNumber, Bet[2], Bet[3]));

Bet[4] = "WIN";

}

else

{

Bet[4] = "DEFEAT";

}

lines[i] = string.Join(" ", Bet);

}

}

using (StreamWriter FileWriter = new StreamWriter(FileUserBets, false))

{

foreach (string UserBet in lines)

{

FileWriter.WriteLine(UserBet);

}

}

}

catch (Exception)

{

return false;

}

return true;

}

}

}

* 1. Вміст файлу SignInWindow.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using BetSystem;

namespace BookmakerOffice

{

public partial class SignInWindow : Form

{

public SignInWindow()

{

InitializeComponent();

}

public void clear\_information()

{

tb\_Login.Text = "";

tb\_Password.Text = "";

}

private void ll\_SignUp\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

clear\_information();

SignUpWindow window = new SignUpWindow();

window.ShowDialog();

}

private void b\_SignIn\_Click(object sender, EventArgs e)

{

IUserInterface User = new SystemManagment();

IAdminInterface Admin = new SystemManagment();

if(User.sign\_in(tb\_Login.Text, tb\_Password.Text))

{

clear\_information();

UserWindow window = new UserWindow(User);

window.ShowDialog();

}

else if(Admin.sign\_in\_admin(tb\_Login.Text, tb\_Password.Text))

{

clear\_information();

AdminWindow window = new AdminWindow(Admin);

window.ShowDialog();

}

else

{

MessageBox.Show("Incorrect login or password", "Error!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

* 1. Вміст файлу SignUpWindow.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using BetSystem;

namespace BookmakerOffice

{

public partial class SignUpWindow : Form

{

public SignUpWindow()

{

InitializeComponent();

}

private void B\_SignUp\_Click(object sender, EventArgs e)

{

if(tb\_Login.Text == "")

{

MessageBox.Show("Please, enter login", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else if(tb\_Password.Text == "" || tb\_ConfirmPassword.Text == "")

{

MessageBox.Show("Please, enter you password", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else if(tb\_Password.Text != tb\_ConfirmPassword.Text)

{

MessageBox.Show("Passwords do not match", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else if(tb\_Login.Text != "" && tb\_Password.Text == tb\_ConfirmPassword.Text && tb\_Password.Text != "")

{

IUserInterface obj = new SystemManagment();

if(obj.sign\_up(tb\_Login.Text, tb\_Password.Text))

{

MessageBox.Show("Account successfully registered", "Information!", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Close();

}

else

{

MessageBox.Show("Account with this login already exists", "Error!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

}

* 1. Вміст файлу UserWindow.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using BetSystem;

namespace BookmakerOffice

{

public partial class UserWindow : Form

{

IUserInterface User;

string CurrentMatchNumber = "";

string CurrentBet = "";

public UserWindow(IUserInterface Father)

{

InitializeComponent();

User = Father;

l\_Money.Text = User.get\_money();

l\_Wins.Text = User.get\_wins();

l\_UserName.Text = User.get\_login();

clear\_interface();

update\_matches();

update\_history();

}

private void pb\_Money\_Click(object sender, EventArgs e)

{

User.plus\_one\_coin();

l\_Money.Text = User.get\_money();

}

private void ll\_SignOut\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

this.Close();

}

private void clear\_bets()

{

CurrentBet = "";

b\_W1.BackColor = Color.Gainsboro;

b\_D.BackColor = Color.Gainsboro;

b\_W2.BackColor = Color.Gainsboro;

b\_T0.BackColor = Color.Gainsboro;

b\_T1.BackColor = Color.Gainsboro;

b\_T2.BackColor = Color.Gainsboro;

b\_T3.BackColor = Color.Gainsboro;

b\_T4Plus.BackColor = Color.Gainsboro;

}

private void clear\_interface()

{

l\_FirstTeam.Text = "Team 1";

l\_SecondTeam.Text = "Team 2";

b\_W1.Text = "W1\r\n1,00";

b\_D.Text = "D\r\n1,00";

b\_W2.Text = "W2\r\n1,00";

b\_T0.Text = "T0\r\n1,00";

b\_T1.Text = "T1\r\n1,00";

b\_T2.Text = "T2\r\n1,00";

b\_T3.Text = "T3\r\n1,00";

b\_T4Plus.Text = "T4+\r\n1,00";

}

private void set\_bet(string FirstTeam, string SecondTeam, string[] Сoefficient)

{

l\_FirstTeam.Text = FirstTeam;

l\_SecondTeam.Text = SecondTeam;

b\_W1.Text = "W1\r\n" + Сoefficient[0];

b\_D.Text = "D\r\n" + Сoefficient[1];

b\_W2.Text = "W2\r\n" + Сoefficient[2];

b\_T0.Text = "T0\r\n" + Сoefficient[3];

b\_T1.Text = "T1\r\n" + Сoefficient[4];

b\_T2.Text = "T2\r\n" + Сoefficient[5];

b\_T3.Text = "T3\r\n" + Сoefficient[6];

b\_T4Plus.Text = "T4+\r\n" + Сoefficient[7];

}

private void update\_matches()

{

clear\_interface();

clear\_bets();

List<string[]> matches = User.get\_matches();

foreach (string[] match in matches)

{

try

{

dgv\_Matches.Rows.Add(match);

}

catch(Exception)

{

continue;

}

}

}

private void update\_history()

{

dgv\_History.Rows.Clear();

List<string[]> History = User.get\_history();

foreach (string[] Line in History)

{

try

{

dgv\_History.Rows.Add(Line);

}

catch (Exception)

{

continue;

}

}

}

private void B\_Bet\_Click(object sender, EventArgs e)

{

try

{

if (nud\_Sum.Value > Convert.ToDecimal(l\_Money.Text)) throw new Exception();

if (CurrentBet == "" || CurrentMatchNumber == "")

{

MessageBox.Show("Select a match and place a bet!", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (User.new\_bet(CurrentMatchNumber, CurrentBet, nud\_Sum.Value.ToString()))

{

MessageBox.Show("Bet accepted", "Information!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

l\_Money.Text = User.get\_money();

clear\_bets();

nud\_Sum.Value = 1;

update\_history();

}

catch(Exception)

{

MessageBox.Show("Not enough money in the account", "Error!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void B\_W1\_Click(object sender, EventArgs e)

{

clear\_bets();

CurrentBet = "W1";

b\_W1.BackColor = Color.GreenYellow;

}

private void B\_D\_Click(object sender, EventArgs e)

{

clear\_bets();

CurrentBet = "D";

b\_D.BackColor = Color.GreenYellow;

}

private void B\_W2\_Click(object sender, EventArgs e)

{

clear\_bets();

CurrentBet = "W2";

b\_W2.BackColor = Color.GreenYellow;

}

private void B\_T0\_Click(object sender, EventArgs e)

{

clear\_bets();

CurrentBet = "T0";

b\_T0.BackColor = Color.GreenYellow;

}

private void B\_T1\_Click(object sender, EventArgs e)

{

clear\_bets();

CurrentBet = "T1";

b\_T1.BackColor = Color.GreenYellow;

}

private void B\_T2\_Click(object sender, EventArgs e)

{

clear\_bets();

CurrentBet = "T2";

b\_T2.BackColor = Color.GreenYellow;

}

private void B\_T3\_Click(object sender, EventArgs e)

{

clear\_bets();

CurrentBet = "T3";

b\_T3.BackColor = Color.GreenYellow;

}

private void B\_T4Plus\_Click(object sender, EventArgs e)

{

clear\_bets();

CurrentBet = "T4Plus";

b\_T4Plus.BackColor = Color.GreenYellow;

}

private void Dgv\_Matches\_SelectionChanged(object sender, EventArgs e)

{

if (dgv\_Matches.SelectedCells.Count == 0)

{

clear\_bets();

clear\_interface();

}

else

{

DataGridViewRow row = dgv\_Matches.Rows[dgv\_Matches.SelectedCells[0].RowIndex];

if (CurrentMatchNumber != row.Cells[0].Value.ToString())

{

CurrentMatchNumber = row.Cells[0].Value.ToString();

CurrentBet = "";

clear\_bets();

set\_bet(row.Cells[2].Value.ToString().Split(' ')[0], row.Cells[2].Value.ToString().Split(' ')[2], User.get\_match\_coefficient(row.Cells[0].Value.ToString()));

}

}

}

}

}

* 1. Вміст файлу AdminWindow.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using BetSystem;

namespace BookmakerOffice

{

public partial class AdminWindow : Form

{

IAdminInterface Admin;

string CurrentMatchNumber = "";

public AdminWindow(IAdminInterface Father)

{

InitializeComponent();

Admin = Father;

update\_matches();

}

private void L\_SignOut\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

this.Close();

}

public void update\_matches()

{

clear\_interface\_update\_match();

dgv\_Matches.Rows.Clear();

List<string[]> matches = Admin.get\_matches();

foreach (string[] match in matches)

{

try

{

dgv\_Matches.Rows.Add(match);

}

catch (Exception)

{

continue;

}

}

}

void set\_teams\_to\_update(string Team1, string Team2)

{

clear\_interface\_update\_match();

l\_FirstTeam.Text = Team1;

l\_SecondTeam.Text = Team2;

}

private void clear\_interface\_update\_match()

{

l\_FirstTeam.Text = "Team 1";

l\_SecondTeam.Text = "Team 2";

nud\_FirstTeamGoals.Value = 0;

nud\_SecondTeamGoals.Value = 0;

CurrentMatchNumber = "";

}

private void clear\_interface\_new\_match()

{

tb\_Team1.Text = "";

tb\_Team2.Text = "";

dtp\_Date.Value = DateTime.Now;

nud\_W1.Value = 1;

nud\_D.Value = 1;

nud\_W2.Value = 1;

nud\_T0.Value = 1;

nud\_T1.Value = 1;

nud\_T2.Value = 1;

nud\_T3.Value = 1;

nud\_T4Plus.Value = 1;

}

private void B\_AddMatch\_Click(object sender, EventArgs e)

{

if(tb\_Team1.Text != "" && tb\_Team2.Text != "")

{

if(nud\_W1.Value > 1 && nud\_D.Value > 1 && nud\_W2.Value > 1)

{

string Coefficient = nud\_W1.Value.ToString() + ' ' + nud\_D.Value.ToString() + ' ' + nud\_W2.Value.ToString() + ' ';

if (nud\_T0.Value > 1 && nud\_T1.Value > 1 && nud\_T2.Value > 1 && nud\_T3.Value > 1 && nud\_T4Plus.Value > 1)

{

Coefficient += nud\_T0.Value.ToString() + ' ' + nud\_T1.Value.ToString() + ' ' + nud\_T2.Value.ToString() + ' ';

Coefficient += nud\_T3.Value.ToString() + ' ' + nud\_T4Plus.Value.ToString();

if (!Admin.add\_match(tb\_Team1.Text, tb\_Team2.Text, dtp\_Date.Value.ToShortDateString(), Coefficient))

{

MessageBox.Show("Something went wrong!", "Error!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

clear\_interface\_new\_match();

MessageBox.Show("Add match accepted", "Information!", MessageBoxButtons.OK, MessageBoxIcon.Information);

update\_matches();

}

return;

}

}

}

MessageBox.Show("All fields are required!", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

private void Dgv\_Matches\_SelectionChanged(object sender, EventArgs e)

{

if (dgv\_Matches.SelectedCells.Count == 0)

{

clear\_interface\_update\_match();

}

else

{

DataGridViewRow row = dgv\_Matches.Rows[dgv\_Matches.SelectedCells[0].RowIndex];

if (CurrentMatchNumber != row.Cells[0].Value.ToString())

{

set\_teams\_to\_update(row.Cells[2].Value.ToString().Split(' ')[0], row.Cells[2].Value.ToString().Split(' ')[2]);

CurrentMatchNumber = row.Cells[0].Value.ToString();

}

}

}

private void B\_Accept\_Click(object sender, EventArgs e)

{

if(CurrentMatchNumber != "")

{

if(Admin.update\_match(CurrentMatchNumber, nud\_FirstTeamGoals.Value, nud\_SecondTeamGoals.Value))

{

update\_matches();

MessageBox.Show("Update match accepted", "Information!", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("Something went wrong!", "Error!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

MessageBox.Show("Choose the match!", "Warning!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

}

* 1. Скріншоти роботи програми

На рисунку 3.1 зображено скріншот екрану авторизації.

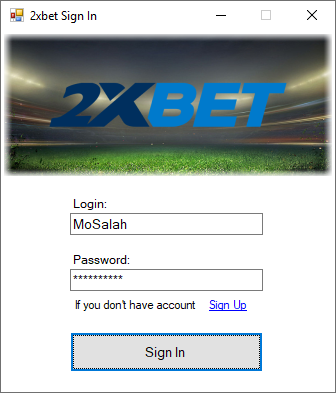


Рисунок 3.1 – Скріншот екрану авторизації

На рисунку 3.2 зображено скріншот екрану реєстрації.

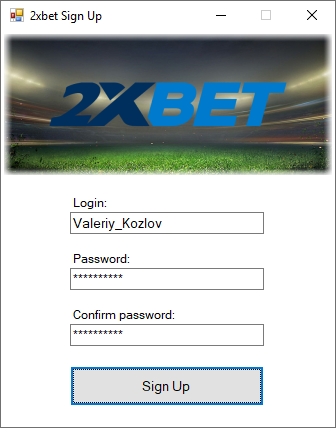


Рисунок 3.2 – Скріншот екрану реєстрації

На рисунку 3.3 зображено головний інтерфейс користувача.

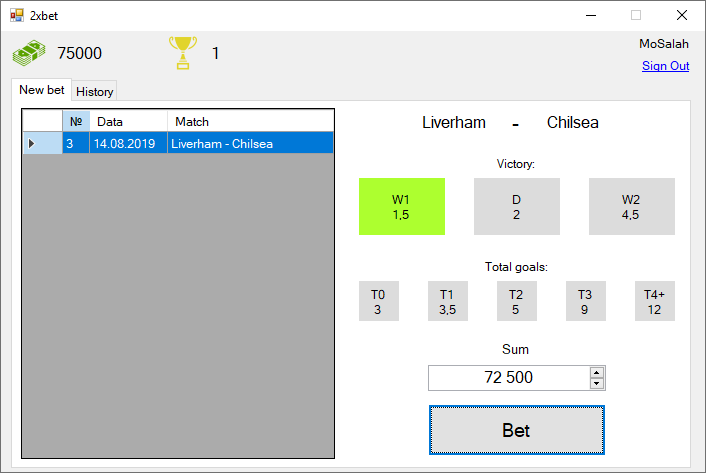


Рисунок 3.3 – Скріншот головного інтерфейсу користувача

На рисунку 3.4 зображено історію ставок користувача.

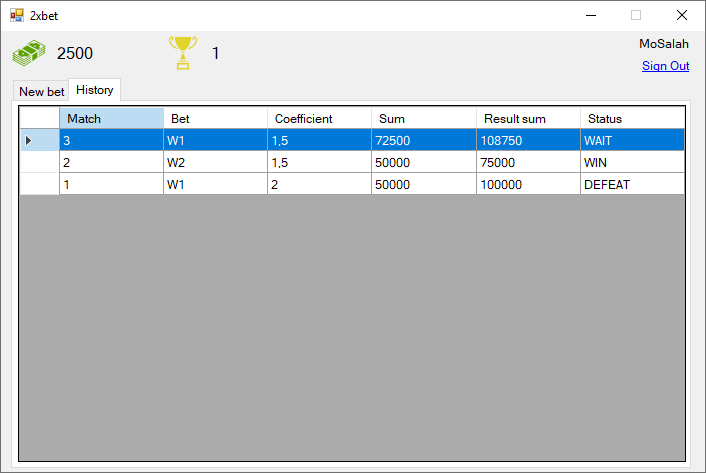


Рисунок 3.4 – Скріншот історії ставок користувача

На рисунку 3.5 зображено головний інтерфейс адміністратора.

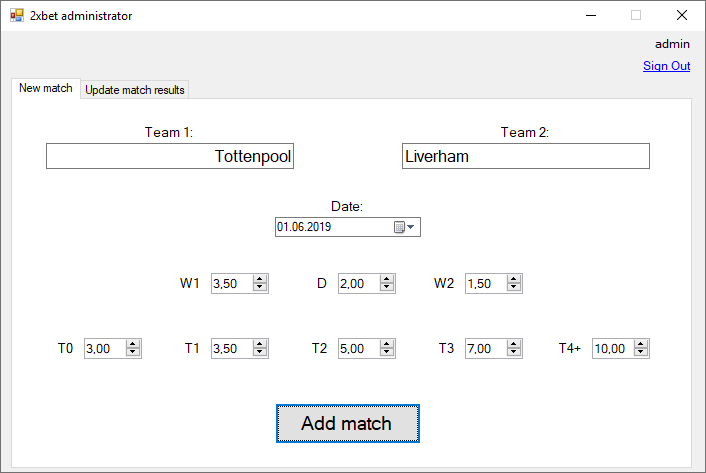


Рисунок 3.5 – Скріншот головного інтерфейсу адміністратора

На рисунку 3.6 зображено зміну результату матчу адміністратором.

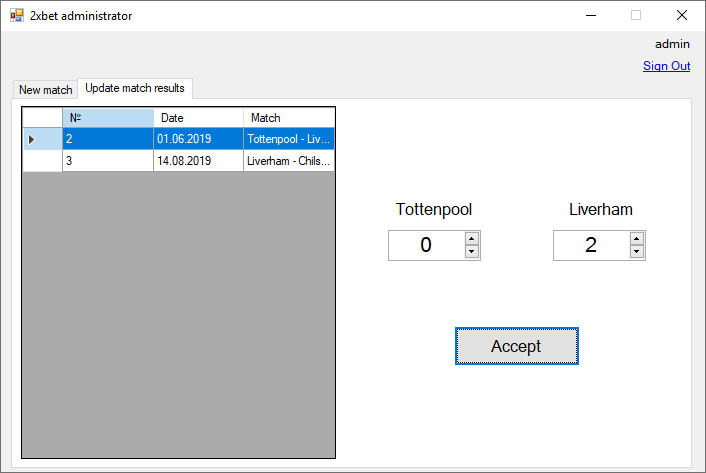


Рисунок 3.6 – Скріншот зміни результату матчу адміністратором

Висновки

Протягом виконання цієї лабораторної роботи ознайомився з принципами компонентно-орієнтованого програмування.

Розробив програмне забезпечення з використанням компонентно-орієнтованого підходу. Навчився використовувати компонентні технології в процесі розроблення програмних систем.