Міністерство освіти і науки України

Національний лісотехнічний університет України

Кафедра інформаційних технологій

**Звіт до лабораторної роботи**

з навчальної дисципліни

**«Методи та засоби ООАП»**

**Виконав:**

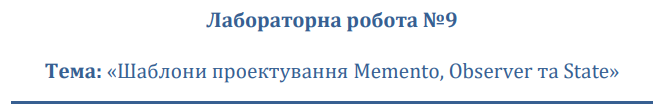
студент групи КНС-21

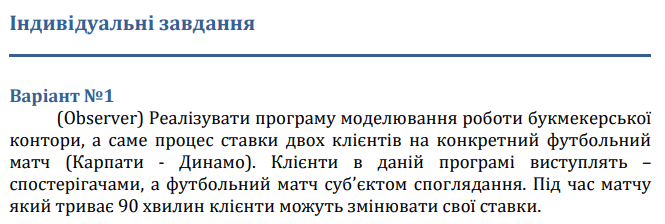
Марущак С. І.

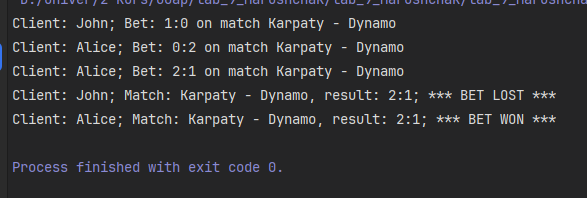
**Перевірив:**

Волинець Є.О.

Львів – 2024







using System;  
using System.Collections.Generic;  
using System.Text.RegularExpressions;  
  
namespace lab\_9\_Marushchak  
{  
public interface IObserver  
 {  
 void Update(AbstractMatch match, string bet);  
 void ReceiveResults(AbstractMatch match, string results);  
   
 }  
  
 public abstract class AbstractMatch  
 {  
 protected List<Subscriber> \_subscribers = new List<Subscriber>();  
 protected string \_match;  
  
 public void Subscribe(IObserver observer, string initialBet)  
 {  
 \_subscribers.Add(new Subscriber(observer, initialBet));  
 }  
  
 public void Unsubscribe(IObserver observer)  
 {  
 \_subscribers.RemoveAll(subscriber => subscriber.Observer == observer);  
 }  
  
 protected void Notify(Subscriber subscriber)  
 {  
 subscriber.Observer.Update(this, subscriber.Bet);  
 }  
   
 public void NotifyResults(string results)  
 {  
 foreach (var subscriber in \_subscribers)  
 {  
 subscriber.Observer.ReceiveResults(this, results);  
 }  
 }  
  
 public abstract string GetMatch();  
 }  
  
 public class FootballMatch : AbstractMatch  
 {  
 public FootballMatch(string match)  
 {  
 \_match = match;  
 }  
  
 public override string GetMatch()  
 {  
 return \_match;  
 }  
  
 public void PlaceBet(Subscriber subscriber, string bet)  
 {  
 subscriber.Bet = bet;  
 Notify(subscriber);  
 }  
   
 public List<Subscriber> GetSubscribers()  
 {  
 return \_subscribers;  
 }  
 }  
  
 public class Client : IObserver  
 {  
 private string \_name;  
 private string \_bet;  
  
 public Client(string name)  
 {  
 \_name = name;  
 }  
  
 public void Update(AbstractMatch match, string bet)  
 {  
 Console.WriteLine($"Client: {\_name}; Bet: {bet} on match {match.GetMatch()}");  
 }  
  
 public void ReceiveResults(AbstractMatch match, string results)  
 {  
   
 if (results == \_bet)  
 {  
 Console.WriteLine($"Client: {\_name}; " +  
 $"Match: {match.GetMatch()}, result: {results}; " +  
 $"\*\*\* BET WON \*\*\*");  
 }  
 else  
 {  
 Console.WriteLine($"Client: {\_name}; " +  
 $"Match: {match.GetMatch()}, result: {results}; " +  
 $"\*\*\* BET LOST \*\*\*");  
 }  
 }  
  
 public void MakeBet(FootballMatch match, string bet)  
 {  
 foreach (var subscriber in match.GetSubscribers())  
 {  
 if (subscriber.Observer == this)  
 {  
 match.PlaceBet(subscriber, bet);  
 \_bet = bet;  
 break;  
 }  
 }  
 }  
 }  
  
 public class Subscriber  
 {  
 public IObserver Observer { get; }  
 public string Bet { get; set; }  
  
 public Subscriber(IObserver observer, string bet)  
 {  
 Observer = observer;  
 Bet = bet;  
 }  
 }  
  
 internal class Program  
 {  
 public static void Main(string[] args)  
 {  
 FootballMatch footballMatch = new FootballMatch("Karpaty - Dynamo");  
 Client client1 = new Client("John");  
 Client client2 = new Client("Alice");  
  
 footballMatch.Subscribe(client1, "0:0");  
 footballMatch.Subscribe(client2, "0:0");  
  
 client1.MakeBet(footballMatch, "1:0");  
 client2.MakeBet(footballMatch, "0:2");  
 client2.MakeBet(footballMatch, "2:1");  
  
   
 footballMatch.NotifyResults("2:1");  
 }  
 }  
}

**Висновок:** на даній лабораторній роботі використовував шаблон проектування Observer.