Національний технічний університет України «КПІ ім. І. Сікорського»

Факультет інформатики і обчислювальної техніки

Кафедра автоматизованих систем обробки інформації та управління

**Лабораторна робота №5**

**з дисципліни**

**«ОРПЗ .NET»**

на тему

# «Шаблони проектування. Поведінкові шаблони»

Виконав

студент групи ІС-03

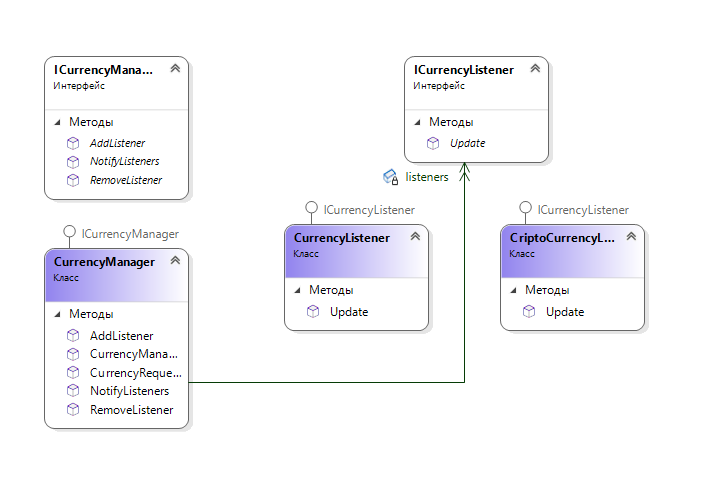
Овсепян Роман

Київ – 2022



Я обрав паттерн Observable, оскільки у мене є декілька класів, яким необхідно відслідковувати дещо спільне(в даному випадку, курс валют)

Діаграма класів



Observable:

using Lab\_5.Listeners;

namespace Lab\_5.Observable

{

public interface ICurrencyManager

{

void AddListener(ICurrencyListener l);

void RemoveListener(ICurrencyListener l);

void NotifyListeners(string data);

}

}

using Lab\_5.Listeners;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_5.Observable

{

public class CurrencyManager : ICurrencyManager

{

private List<ICurrencyListener> listeners;

public CurrencyManager()

{

listeners = new List<ICurrencyListener>();

}

public void AddListener(ICurrencyListener l)

{

listeners.Add(l);

}

public void RemoveListener(ICurrencyListener l)

{

listeners.Remove(l);

}

public void NotifyListeners(string data)

{

foreach(ICurrencyListener l in listeners)

{

l.Update(data);

}

}

public async Task CurrencyRequestAsync()

{

HttpWebRequest request = (HttpWebRequest)WebRequest

.Create("https://api.privatbank.ua/p24api/pubinfo?exchange&coursid=5");

HttpWebResponse response = (HttpWebResponse)await request.GetResponseAsync();

Stream receiveStream = response.GetResponseStream();

StreamReader readStream = new StreamReader(receiveStream, Encoding.UTF8);

var result = readStream.ReadToEnd();

response.Close();

NotifyListeners(result);

}

}

}

Listeners:

namespace Lab\_5.Listeners

{

public interface ICurrencyListener

{

void Update(string data);

}

}

using System;

using System.Linq;

using System.Xml.Linq;

namespace Lab\_5.Listeners

{

public class CurrencyListener : ICurrencyListener

{

public void Update(string data)

{

XDocument doc = XDocument.Parse(data);

XElement currency = doc.Element("exchangerates");

if(currency != null)

{

var res = currency.Descendants("row")

.Where(x => x.Element("exchangerate").Attribute("ccy").Value != "BTC")

.Select(x => new {

CType = x.Element("exchangerate").Attribute("ccy").Value,

CBuy = x.Element("exchangerate").Attribute("buy").Value,

CSale = x.Element("exchangerate").Attribute("buy").Value,

});

foreach(var item in res)

{

Console.WriteLine($"CCY: {item.CType}, Buy: {item.CBuy}, Sale: {item.CSale}");

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Xml.Linq;

namespace Lab\_5.Listeners

{

public class CriptoCurrencyListener : ICurrencyListener

{

public void Update(string data)

{

XDocument doc = XDocument.Parse(data);

XElement currency = doc.Element("exchangerates");

if (currency != null)

{

var res = currency.Descendants("row")

.Where(x => x.Element("exchangerate").Attribute("ccy").Value == "BTC")

.Select(x => new {

CType = x.Element("exchangerate").Attribute("ccy").Value,

CBuy = x.Element("exchangerate").Attribute("buy").Value,

CSale = x.Element("exchangerate").Attribute("buy").Value,

});

foreach (var item in res)

{

Console.WriteLine($"CCY: {item.CType}, Buy: {item.CBuy}, Sale: {item.CSale}");

}

}

}

}

}

Application:

using Lab\_5.Listeners;

using Lab\_5.Observable;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

using System.Xml;

using System.Xml.Linq;

namespace Lab\_5

{

public class Program

{

static async Task Main(string[] args)

{

CurrencyManager currencyManager = new CurrencyManager();

CurrencyListener currencyListener = new CurrencyListener();

CriptoCurrencyListener criptoCurrencyListener = new CriptoCurrencyListener();

currencyManager.AddListener(currencyListener);

currencyManager.AddListener(criptoCurrencyListener);

bool carryOn = true;

while (carryOn)

{

Console.Write("Press Enter to get currency info(Type 'stop' to quit): ");

string answer = Console.ReadLine();

if(answer.ToLower() == "stop")

{

break;

}

await currencyManager.CurrencyRequestAsync();

Console.WriteLine();

}

}

}

}