МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Національний технічний університет України

Факультет інформатики та обчислювальної техніки

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра інформаційних систем та технологій

Звіт

з лабораторної  роботи № 2

з дисципліни «Програмування – 2.

Основи програмування»

Варіант № 5

Виконав: Валіваха Андрій

     Студент гр. ІС-11 , ФІОТ

1 курс

Київ 2022

**Лабораторна робота #2**

Поліморфізм. Наслідування. Перегрузка методів

**Завдання 1**

Створити додаток, який задовольняє вимогам, наведеним в завданні. Наслідування застосовувати тільки в тих завданнях, в яких воно **логічно обґрунтоване.** Аргументувати належність класу кожного створюваного методу і коректно перевизначити для кожного класу методи *Equals*, *GetHashCode*, *ToString*. При виклику будь-которого методу класу, виводити на екран текстове повідомлення.

using System;

namespace Lab\_OP\_2

{

class Animal

{

public string Kind { get; }

public string Name { get; }

public Animal(string kind, string name)

{

Kind = kind;

Name = name;

}

public override bool Equals(object obj)

{

if (obj is Animal animal) return Kind == animal.Kind && animal.Name == Name;

else return false;

}

public override string ToString()

{

return $"Pet {Kind} {Name}";

}

public override int GetHashCode()

{

return (Kind, Name).GetHashCode();

}

}

class Dog : Animal

{

public string Breed { get; }

public Dog(string kind, string name, string breed) : base(kind, name)

{

Breed = breed;

}

public override bool Equals(object obj)

{

if (obj is Dog dog)

return Breed == dog.Breed;

return false;

}

public override string ToString()

{

return $"Dog {Breed}";

}

public override int GetHashCode()

{

return Breed.GetHashCode();

}

public virtual void MakeSound()

{

Console.WriteLine("ruff");

}

public void Bite()

{

Console.WriteLine("ouuch!!!");

}

public void Jump()

{

Console.WriteLine("\*Jumps\*");

}

}

class Puppy : Dog

{

public int Age { get; }

public Puppy(int age, string kind, string name, string breed) : base(kind, name, breed)

{

Age = age;

}

public override void MakeSound()

{

Console.WriteLine("Wooof");

}

public override bool Equals(object obj)

{

if (obj is Puppy puppy)

return Name == puppy.Name && puppy.Age == Age;

return false;

}

public override string ToString()

{

return $"Puppy {Age}\n {Name}\n {Kind}\n {Breed}\n";

}

public override int GetHashCode()

{

return (Name, Age).GetHashCode();

}

static void Main(string[] args)

{

var puppy = new Puppy(5, "Name: Joystick", "Kind : Sheepdog ", "Breed: Corgi");

Console.WriteLine(puppy);

puppy.MakeSound();

puppy.Bite();

puppy.Jump();

}

}

}

**Завдання 2.**

Створити програму, яка задовольняє наступним вимогам:

• Використовувати можливості ООП: класи, наслідування, поліморфізм, інкапсуляція.

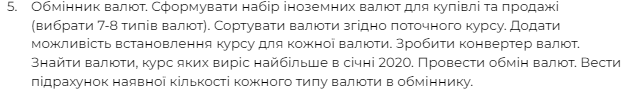
• Кожен клас повинен мати змістовну назву та інформативний склад.

• Наслідування має застосовуватися тільки тоді, коли це має сенс.

• Класи повинні бути грамотно розкладені по пакетах.

• Консольне меню повинно бути мінімальним.

• Для зберігання параметрів ініціалізації можна використовувати файли.



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;

namespace OPLab\_2\_2ndTask

{

public partial class Form1 : Form

{

Dictionary<string, List<double>> currencies = new Dictionary<string, List<double>>()

{

{ "USD", new List<double>(){ 1, 1.1, 0.9 } },

{ "UAH", new List<double>(){ 8, 25, 27, 28, 34 } },

{ "EUR", new List<double>(){ 0.7, 0.74, 0.8, 0.94 } },

{ "PLN", new List<double>(){ 3.78, 3.60, 4.4, 4.32 } },

{ "GBP", new List<double>(){ 0.6, 0.78, 0.8 } },

{ "CHF", new List<double>(){ 0.87, 0.96 } },

{ "AED", new List<double>(){ 3.3, 3.47, 3.67 } }

};

public Form1()

{

InitializeComponent();

fromcombo1.Items.Clear();

tocombo2.Items.Clear();

List<(string, double)> lastCurrencies = new List<(string, double)>();

foreach (var item in currencies)

{

lastCurrencies.Add((item.Key, item.Value.Last()));

}

lastCurrencies = lastCurrencies.OrderBy(i => i.Item2).ToList();

foreach (var currency in lastCurrencies)

{

fromcombo1.Items.Add(currency.Item1);

tocombo2.Items.Add(currency.Item1);

}

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

int i = int.Parse(amount\_txt.Text);

string fromCurrency = fromcombo1.SelectedItem.ToString();

string toCurrency = tocombo2.SelectedItem.ToString();

double fromKoef = currencies[fromCurrency].Last();

double toKoef = currencies[toCurrency].Last();

double conver = i \* toKoef / fromKoef;

display\_txt.Text = "Converted Amount :" + conver + "\t " + toCurrency;

CurrRate.Text = "Currency Exchange Rate: " + (toKoef / fromKoef).ToString();

}

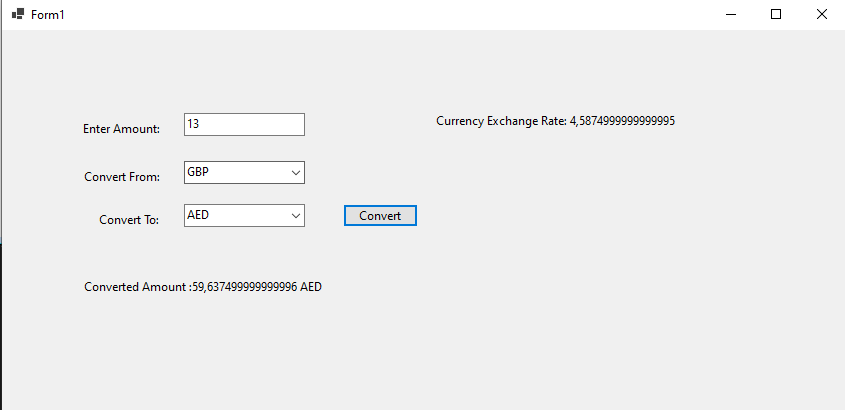
private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

}



Висновок: Виконавши дану лабораторну роботу, я навчився використовувати, поліморфізм, наслідування та перегрузку методів.