Міністерство освіти та науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

ЗВІТ

до лабораторної роботи №1

з дисципліни «Основи Web-програмування»

Виконав: студент 2 курсу групи ІП-64 , ФІОТ

Євтушенко Ілля

Номер залікової: 6406 Номер в списку: 7

Київ – 2017

**Код програми:**

using System;

using System.Collections.Generic;

namespace seashark2

{

class Program

{

public abstract class Currency

{

protected double Money { get; set; }

public abstract Currency ToDollar();

public abstract Currency ToEuro();

public abstract Currency ToPound();

public abstract Currency ToHryvnya();

}

public class Dollar : Currency

{

public double number

{

get

{

return Money;

}

set

{

Money = value;

}

}

public override Currency ToDollar()

{

Dollar newvalue = new Dollar();

newvalue.number = this.number;

return newvalue;

}

public override Currency ToEuro()

{

Dollar newvalue = new Dollar();

newvalue.number = this.number \* 1.2;

return newvalue;

}

public override Currency ToHryvnya()

{

Dollar newvalue = new Dollar();

newvalue.number = this.number \* 27;

return newvalue;

}

public override Currency ToPound()

{

Dollar newvalue = new Dollar();

newvalue.number = this.number \* 2;

return newvalue;

}

public override string ToString()

{

return string.Format("{0}", this.number);

}

}

public class Pounds : Currency

{

public double number

{

get

{

return Money;

}

set

{

Money = value;

}

}

public override Currency ToDollar()

{

Pounds newvalue = new Pounds();

newvalue.number = this.number/2;

return newvalue;

}

public override Currency ToEuro()

{

Pounds newvalue = new Pounds();

newvalue.number = this.number/1.8;

return newvalue;

}

public override Currency ToHryvnya()

{

Pounds newvalue = new Pounds();

newvalue.number = this.number \* 40;

return newvalue;

}

public override Currency ToPound()

{

Dollar newvalue = new Dollar();

newvalue.number = this.number;

return newvalue;

}

public override string ToString()

{

return string.Format("{0}", this.number);

}

}

public class Euro : Currency

{

public double number

{

get

{

return Money;

}

set

{

Money = value;

}

}

public override Currency ToDollar()

{

Euro newvalue = new Euro();

newvalue.number = this.number/1.2;

return newvalue;

}

public override Currency ToEuro()

{

Euro newvalue = new Euro();

newvalue.number = this.number;

return newvalue;

}

public override Currency ToHryvnya()

{

Euro newvalue = new Euro();

newvalue.number = this.number \* 35;

return newvalue;

}

public override Currency ToPound()

{

Euro newvalue = new Euro();

newvalue.number = this.number \* 2;

return newvalue;

}

public override string ToString()

{

return string.Format("{0}", this.number);

}

}

public class Purse : List<Currency>

{

private int size;

public Currency[] array;

private int index = 0;

public Purse(int sized)

{

size = sized;

array = new Currency[size];

}

public new Boolean Add (Currency x) {

array[index] = x;

index++;

return true;

}

public int length

{

get

{

return array.Length;

}

}

public override string ToString()

{

string vivod = "";

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

{

vivod += string.Format(i + 1 + ". {0} \n", this.array[i]);

}

return vivod;

}

public override bool Equals(object obj)

{

if (!(obj is Purse))

{

return false;

}

else

{

return ((Purse)obj).array == this.array;

}

}

public static bool operator == (Purse one, Purse two)

{

if ((Object)one == null || (Object)two == null)//проверить на null

return false;

return one.Equals(two);

}

public static bool operator != (Purse one, Purse two)

{

if ((Object)one == null || (Object)two == null)//проверить на null

return true;

return !one.Equals(two);

}

public override int GetHashCode()

{

try

{

return Convert.ToInt32(this.array);

}

catch (Exception e)

{

Console.WriteLine(e.Message);

return -1;

}

}

public Purse DeepCopy()

{

Purse obj = new Purse(size);

return obj;

}

class MyInvalidCastException : InvalidCastException

{

public MyInvalidCastException()

: base() { }

public MyInvalidCastException(string message)

: base(message) { }

public MyInvalidCastException(string format, params object[] args)

: base(string.Format(format, args)) { }

public MyInvalidCastException(string message, Exception innerException)

: base(message, innerException) { }

public MyInvalidCastException(string format, Exception innerException, params object[] args)

: base(string.Format(format, args), innerException) { }

}

}

static void Main(string[] args)

{

Console.WriteLine("Вот лаба");

Dollar first = new Dollar();

first.number = 80.76;

Euro second = new Euro();

second.number = 36.36;

Pounds third = new Pounds();

third.number = 88.88;

Console.WriteLine("Доллары: " + first.number);

Console.WriteLine("Доллары в гривну: " + (first.ToHryvnya()).ToString());

Console.WriteLine("В доллары обратно: " + first.number);

Console.WriteLine("Евро: " + second.number);

Console.WriteLine("Евро в гривну: " + (second.ToHryvnya()).ToString());

Console.WriteLine("фунты: " + third.number);

Console.WriteLine("Фунты в гривну: " + (third.ToHryvnya()).ToString());

Purse Gamanets = new Purse(3);

Gamanets.Add(first);

Gamanets.Add(second);

Gamanets.Add(third);

Console.WriteLine("Мои деньги в гаманце:\n" + Gamanets.ToString());

}

}

}

**Скріншот програми:**

