Khmelnytskyi National University

Department of Computer Engineering and Information Systems

**Report**

Laboratory work №2

Discipline: “Object-oriented programming”

Topic: “Syntactic features of the C# programming language

Classes and structures. Constructors and destructors”

Completed: 1st year student, group CE-23-1 Evgen Gulevaty

Name, Surname

Checked: BoikoV.O.

Name, Surname

Khmelnytskyi, 2024

Purpose: Learn the basic principles of inheritance in C#. Learn to write programs using the inheritance mechanism. Gain skills in using the polymorphism mechanism.

**Task 1**

Write a C# program that implements the concept of inheritance Keywords new, base, sealed, this, override Abstract and virtual methods Sealed classes and class members Polymorphism: In a hairdressing salon, the Manager accepts orders from customers and forms queues for hairdressers. The hairdresser serves the client, for which he receives money and provides a catalog of possible hairstyles. The client makes an appointment, chooses a haircut from the catalog and pays for the haircut.

**1.2. Program text**

using MaterialSkin;

using MaterialSkin.Controls;

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 static System.Windows.Forms.VisualStyles.VisualStyleElement.Button;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Lab2\_Gulevaty

{

public partial class Form1 : MaterialForm

{

private int clientId = 0;

private List<Client> clientList = new List<Client>();

private Dictionary<int, int> haircuts = new Dictionary<int, int>();

public Form1()

{

InitializeComponent();

}

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

{

string name = textBox1.Text;

DateTime date = dateTimePicker1.Value;

if (!radioButton1.Checked && !radioButton2.Checked &&

!radioButton3.Checked && !radioButton4.Checked)

{

MessageBox.Show("Choose your haircut");

return;

}

int haircutCost = 0;

if (radioButton1.Checked)

{

haircutCost = 200;

}

else if (radioButton2.Checked)

{

haircutCost = 150;

}

else if (radioButton3.Checked)

{

haircutCost = 300;

}

else if (radioButton4.Checked)

{

haircutCost = 5;

}

Client client = new Client(name, date, clientId++, haircutCost);

clientList.Add(client);

haircuts.Add(client.ClientId, haircutCost);

radioButton1.Checked = false;

radioButton2.Checked = false;

radioButton3.Checked = false;

radioButton4.Checked = false;

textBox1.Text = "";

Form2 formReceiver = new Form2(clientList, haircuts);

formReceiver.Show();

}

}

}

public class Person

{

public string Name { get; set; }

public DateTime Date { get; set; }

public Person(string name, DateTime date)

{

Name = name;

Date = date;

}

public override string ToString()

{

return $"{Name} - {Date}";

}

}

sealed public class Client : Person

{

public int ClientId { get; set; }

public int HaircutCost { get; set; }

public Client(string name, DateTime date, int clientId, int haircutCost)

: base(name, date)

{

ClientId = clientId;

HaircutCost = haircutCost;

}

}

}

//Form2  
using MaterialSkin;

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 static System.Windows.Forms.VisualStyles.VisualStyleElement;

using System.IO;

namespace Lab2\_Gulevaty

{

public partial class Form2 : Form

{

private List<Client> clientList;

private Dictionary<int, int> haircuts;

private int totalMoney = 0;

private int chek = 0;

public Form2(List<Client> clients, Dictionary<int, int> haircutPrices)

{

InitializeComponent();

clientList = clients;

haircuts = haircutPrices;

if (clientList != null)

{

foreach (var client in clientList)

{

listBox1.Items.Add(client);

}

using (var stream = File.OpenWrite("data.txt"))

using (var writer = new StreamWriter(stream))

{

foreach (var item in clientList)

{

writer.WriteLine($"{item.Name},{item.Date},{" --- "},{item.ClientId}");

}

}

}

}

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

{

if (listBox1.SelectedIndex != -1)

{

var selectedClient = clientList[listBox1.SelectedIndex];

int cost;

if (haircuts.TryGetValue(selectedClient.ClientId, out cost))

{

totalMoney += cost;

}

using (var stream = File.OpenWrite("data.txt"))

using (var writer = new StreamWriter(stream))

{

writer.WriteLine($"{totalMoney}");

}

textBox1.Text = "";

textBox1.Text = totalMoney.ToString();

clientList.RemoveAt(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(listBox1.SelectedIndex);

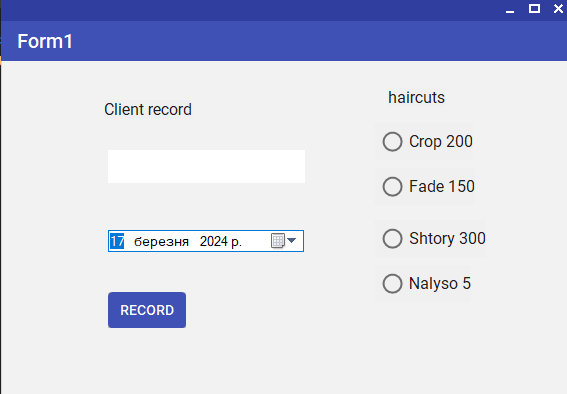
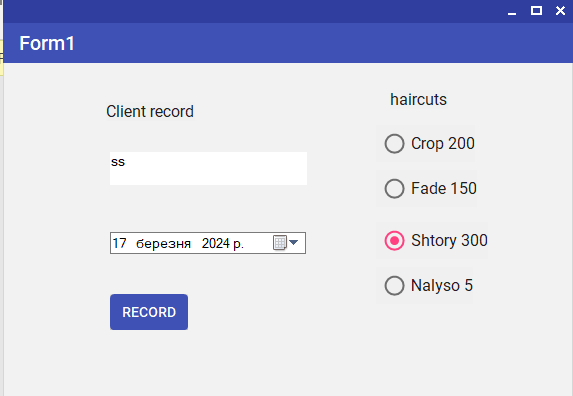
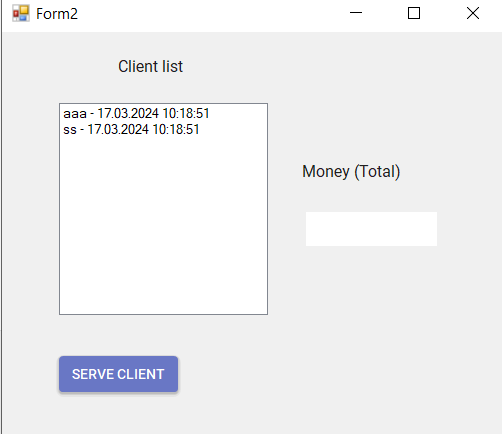
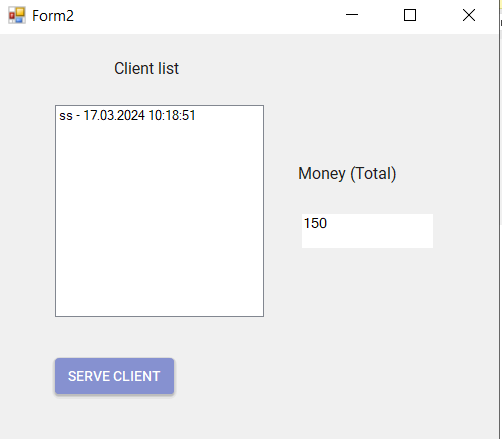
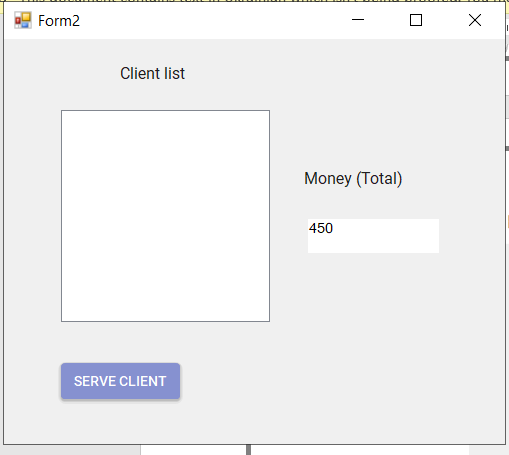
chek++;

}

}

}

}

**1.3. Program results:**  ****  
****  
****  
****  
****