МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ПОЛТАВСЬКА ПОЛІТЕХНІКА

ІМЕНІ ЮРІЯ КОНДРАТЮКА»

Навчально науковий інститут інформаційних технологій і робототехніки

Кафедра комп’ютерних та інформаційних технологій і систем

**Лабораторна робота № 2**

з навчальної дисципліни

"ОБ’ЄКТНО ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ"

Варіант – 8

**Виконала:**

*студента 201-ТК*

*Зуєва Микита Сергійовича*

**Перевірив:**

*Демиденко Максим Ігорович*

Полтава 2023

**Індивідуальне завдання**

Побудувати опис класу, що містить інформацію про поштову адресу організації. Передбачити можливість роздільної зміни складових частин адреси, створення й знищення об’єктів цього класу.

Код програми:

Form1.cs

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

namespace Lab2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private List<PostalData> postalDataList = new List<PostalData>();// створення списку для збереження об'єктів

// клік по кнопці ОК

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

{

//створення об'єкта

PostalData postalData = new PostalData(txtInputIndex.Text, txtInputAddress.Text, txtInputName.Text, txtInputEmail.Text, txtInputPhoneNumber.Text);

postalDataList.Add(postalData); // додаємо отримані дані до postalDataList

listBoxOutput.Items.Add(postalData.ToString()); // додаємо дані на listBoxOutput, для відображення на екрані

}

// клік по кнопці Update

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

{

if (listBoxOutput.SelectedIndex != -1) // перевірка чи вибрано елемент у listBoxOutput

{

int index = listBoxOutput.SelectedIndex; //отримання індексу обраного елементу

PostalData selectedData = postalDataList[index]; //отримання відповідного об'єкту у списку postalDataList

// використання створеного методу UpdatePostalData, для оновлення даних

selectedData.UpdatePostalData(txtInputIndex.Text, txtInputAddress.Text, txtInputName.Text, txtInputEmail.Text,txtInputPhoneNumber.Text);

listBoxOutput.Items[index] = selectedData.ToString(); //оновлення обраного елемента

}

}

//клік по кнопці Delete

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

{

if (listBoxOutput.SelectedIndex != -1) // перевірка чи вибрано елемент у listBoxOutput

{

int index = listBoxOutput.SelectedIndex; //отримання індексу обраного елементу

postalDataList.RemoveAt(index); //видаляємо елемент з списку postalDataList

listBoxOutput.Items.RemoveAt(index); ////видаляємо елемент з listBoxOutput

}

}

// натискання кнопки в полі редагування

private void txtInputIndex\_KeyPress(object sender, KeyPressEventArgs e)

{

if (char.IsDigit(e.KeyChar)) //перевірка чи вводяться цифри

{

if (txtInputIndex.Text.Length < 6) // дозволено ввести тільки 6 цифр

{

return;

}

else

{

e.Handled = true; // якщо користувач намагається ввести більше 6 цифр, введення заборонено

}

}

else if (char.IsControl(e.KeyChar)) // перевірка чи вводяться <Enter>, <Backspace>, <Esc>

{

// <Enter>, <Backspace>, <Esc>

btnOK.Focus();

return;

}

else

{

// інші символи заборонені

e.Handled = true;

}

}

}

}

UsersPostalData.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab2

{

internal class PostalData

{

// поля класу

private string index;

private string address;

private string name;

private string email;

private string phoneNumber;

// властивість для доступу до поля index

public string Index

{

get { return index; }

set { index = value; }

}

// властивість для доступу до address

public string Address

{

get { return address; }

set { address = value; }

}

// властивість для доступу до name

public string Name

{

get { return name; }

set { name = value; }

}

// властивість для доступу до email

public string Email

{

get { return email; }

set { email = value; }

}

// властивість для доступу до phoneNumber

public string PhoneNumber

{

get { return phoneNumber; }

set { phoneNumber = value; }

}

//конструктор без параметрів

public PostalData()

{

index = "000000";

address = "";

name = "";

email = "";

phoneNumber = "";

}

//конструктор з двома параметрами

public PostalData(string index, string address)

{

this.index = index;

this.address = address;

name = "";

email = "";

phoneNumber = "";

}

//конструктор з троьма параметрами

public PostalData(string index, string address, string name)

{

this.index = index;

this.address = address;

this.name = name;

email = "";

phoneNumber = "";

}

//конструктор з чотирма параметрами

public PostalData(string index, string address, string name, string email)

{

this.index = index;

this.address = address;

this.name = name;

this.email = email;

phoneNumber = "";

}

//конструктор з п'ятьма парамтерами

public PostalData(string index, string address, string name, string email, string phoneNumber)

{

this.index = index;

this.address = address;

this.name = name;

this.email = email;

this.phoneNumber = phoneNumber;

}

//перевантаження методу ToString()

public override string ToString()

{

return $"Індекс: {Index} Адреса: {Address} Прізвище та ім'я: {Name} Email: {Email} Телефон: {PhoneNumber}";

}

//метод для оновлення даних

public void UpdatePostalData(string newIndex, string newAddress, string newName, string newEmail, string newPhoneNumber)

{

Index = newIndex;

Address = newAddress;

Name = newName;

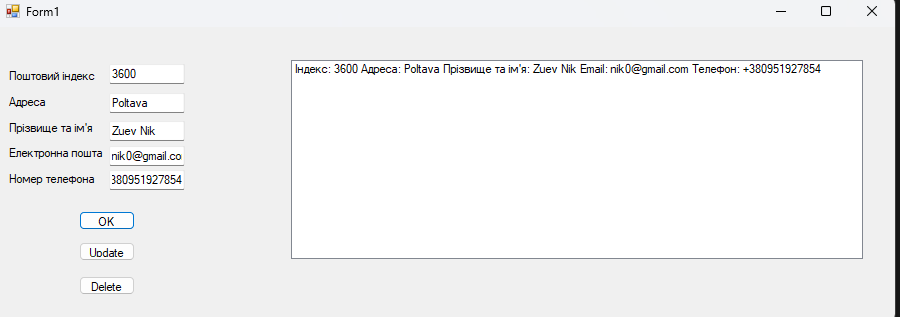
Email = newEmail;

PhoneNumber = newPhoneNumber;

}

}

}



Посилання на файли проєкту:

https://github.com/Nikita-Zuev/OOP