



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

О Т Ч Е Т

по лабораторной работе № 4

Название: Windows Forms

Дисциплина: Разработка приложений на языке C#

Студент

ИУ6-75Б

(Группа)

(Подпись, дата)

И.Ю. Жосан

(И.О. Фамилия)

Преподаватель

А.М. Минитаева

(Подпись, дата)

(И.О. Фамилия)

Москва, 2023

Цель работы: изучить основы разработки интерфейса программы с помощью Windows Forms.

Задание: создать интерфейс к программе, созданной в лабораторной работе 3. Ввод сведений о новых клиентах и счетах реализовать через дополнительные формы.

Выполнение задания:

Для выполнения поставленной задачи были созданы 4 формы:

- начальное меню (рисунок 1);
- форма для ввода данных о клиенте при создании или поиске (рисунок 2);
- форма создания или выбора банковского счета (рисунок 3);
- меню возможностей работы со счетом (рисунок 4).

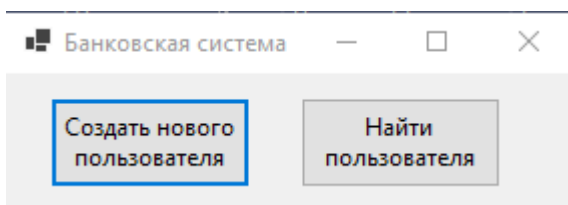


Рисунок 1 – Начальное меню

Рисунок 2 – Форма ввода данных о клиенте

Рисунок 3 – Форма создания или выбора банковского счета

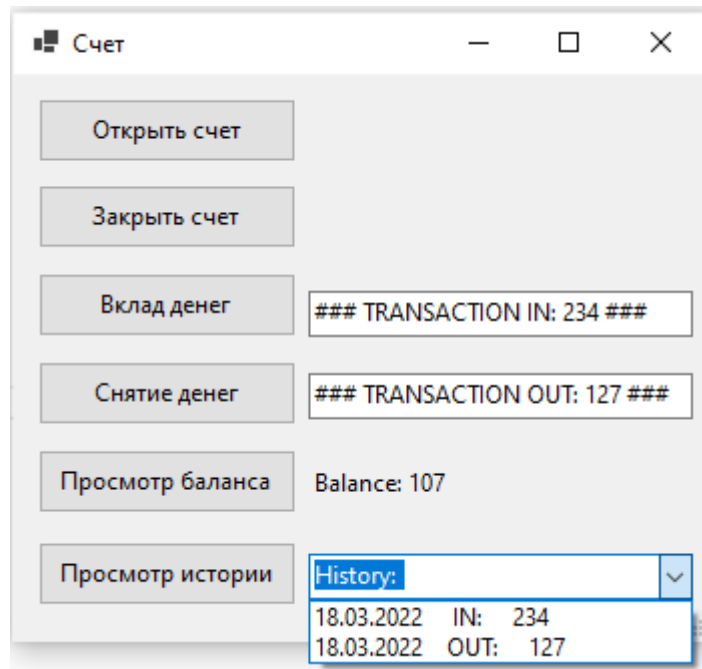


Рисунок 4 – Меню возможностей работы со счетом

Код основной программы:

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace lab4_windowsApp
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.SetHighDpiMode(HighDpiMode.SystemAware);
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new startForm());
        }
    }

    public class Client
    {
        private string first_name, last_name, father_name, job;
        private int year_of_birth;
        private int[] bank_accounts;
        public Client(string firstName, string lastName, string
fatherName, int yearOfBirth, string job_)
        {
            first_name = firstName;
            last_name = lastName;
            father_name = fatherName;
        }
    }
}
```

```

        year_of_birth = yearOfBirth;
        job = job_;
        bank_accounts = new int[0];
    }
    public override string ToString()
    {
        return first_name + ' ' + last_name + ' ' + father_name
+ ' ' + year_of_birth.ToString();
    }
    public void addAccount(int num)
    {
        Array.Resize(ref bank_accounts, bank_accounts.Length +
1);

        bank_accounts[bank_accounts.Length - 1] = num;
    }
    public int[] GetAccounts() { return bank_accounts; }
}
public class Transaction
{
    private double sum;
    private string date;
    private bool direction;
    public Transaction(double s, string d, bool dir)
    {
        sum = s;
        date = d;
        direction = dir;    // 1 - in    0 - out
    }
    public override string ToString()
    {
        return direction ? date + "    IN:    " +
sum.ToString() : date + "    OUT:    " + sum.ToString();
    }
}
public class Bank_Account
{
    private static int NumbersForAccounts = 0;
    private int number_of_account;
    private double balance;
    private bool state;
    private Transaction[] history_in, history_out, history;
    public Bank_Account()
    {
        number_of_account = NumbersForAccounts;
        NumbersForAccounts += 1;
        balance = 0.0;
        state = true;
        history_in = new Transaction[0];
        history_out = new Transaction[0];
        history = new Transaction[0];
    }
    public bool GetState() { return state; }
    public void OpenAccount()
    {

```

```

        state = true;
    }
    public void CloseAccount()
    {
        state = false;
    }
    public int GetNumber() { return number_of_account; }
    public void in_money(double sum)
    {
        Transaction t = new Transaction(sum, "18.03.2022",
true);
        Array.Resize(ref history_in, history_in.Length + 1);
        history_in[history_in.Length - 1] = t;
        Array.Resize(ref history, history.Length + 1);
        history[history.Length - 1] = t;
        balance += sum;
    }
    public void out_money(double sum)
    {
        Transaction t = new Transaction(sum, "18.03.2022",
false);
        Array.Resize(ref history_out, history_out.Length + 1);
        history_out[history_out.Length - 1] = t;
        Array.Resize(ref history, history.Length + 1);
        history[history.Length - 1] = t;
        balance -= sum;
    }
    public double getBalance() { return balance; }
    public List<string> getHistory()
    {
        List<string> result = new List<string>();
        for (int i = 0; i < history.Length; i++)
            result.Add(history[i].ToString());
        return result;
    }
}
}

```

Код первой формы:

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace lab4_windowsApp
{
    public partial class startForm : Form
    {
        public static List<Bank_Account> accounts = new
List<Bank_Account>();
        public static List<Client> clients = new List<Client>();
        public startForm()
        {
            InitializeComponent();
        }
    }
}

```

```

        private void newClientCreation_Click(object sender,
EventArgs e)
        {
            InputClientForm f = new InputClientForm();
            f.ShowDialog();
        }
        private void oldClientFinding_Click(object sender, EventArgs
e)
        {
            InputClientForm f = new InputClientForm();
            f.convertToFindForm();
            f.ShowDialog();
        }
    }
}

```

Код второй формы:

```

using System;
using System.Windows.Forms;

namespace lab4_windowsApp
{
    public partial class InputClientForm : Form
    {
        public static string empty_error_str = "### EMPTY ###";
        public static string incorrect_num_error_str = "### NOT NUM
###";
        static string clientDidntExists = "### CLIENT DIDN'T EXIST
###";
        static string clientHasFounded = "### CLIENT HAS FOUNDED
###";
        static string clientHasCreated = "### CLIENT HAS CREATED
###";

        public static Client curClient = null;
        public InputClientForm()
        {
            InitializeComponent();
        }
        private void createButton_Click(object sender, EventArgs e)
        {
            string firstName = this.firstName_textBox.Text;
            string lastName = this.lastName_textBox.Text;
            string fatherName = this.fatherName_textBox.Text;
            string yearSTR = this.year_textBox.Text;
            int year = 0;
            string job = this.job_textBox.Text;
            bool correctSignal = true;
            if (firstName == "")
            {
                this.firstName_textBox.Text = empty_error_str;
                correctSignal = false;
            }
            if (lastName == "")
            {

```

```

        this.lastName_textBox.Text = empty_error_str;
        correctSignal = false;
    }
    if (fatherName == "")
    {
        this.fatherName_textBox.Text = empty_error_str;
        correctSignal = false;
    }
    if (yearSTR == "")
    {
        this.year_textBox.Text = empty_error_str;
        correctSignal = false;
    }
    else
    {
        if (!(int.TryParse(yearSTR, out year)))
        {
            this.year_textBox.Text =
incorrect_num_error_str;
            correctSignal = false;
        }
    }

    if (this.state == true) // create
    {
        if (job == "")
        {
            this.job_textBox.Text = empty_error_str;
            correctSignal = false;
        }
        if (correctSignal)
        {
            Client cl = new Client(firstName, lastName,
fatherName, year, job);
            startForm.clients.Add(cl);
            curClient = cl;
            this.firstName_textBox.Text = clientHasCreated;
            accountForm f = new accountForm();
            f.ShowDialog();
        }
    }
    else // find
    {
        if (correctSignal)
        {
            int i;
            for (i = 0; i < startForm.clients.Count; i++)
            {
                if (startForm.clients[i].ToString() ==
(firstName + ' ' + lastName + ' ' + fatherName + ' ' +
year.ToString()))
                    break;
            }
            if (i == startForm.clients.Count)

```

```

        this.firstName_textBox.Text =
clientDidntExists;
    else
    {
        this.firstName_textBox.Text =
clientHasFounded;
        curClient = startForm.clients[i];
        accountForm f = new accountForm();
        f.ShowDialog();
    }
}

private void backButton_Click(object sender, EventArgs e)
{
    this.Close();
}

private void firstName_textBox_Click(object sender,
EventArgs e)
{
    if (this.firstName_textBox.Text == empty_error_str)
        this.firstName_textBox.Clear();
}

private void lastName_textBox_Click(object sender, EventArgs
e)
{
    if (this.lastName_textBox.Text == empty_error_str)
        this.lastName_textBox.Clear();
}

private void fatherName_textBox_Click(object sender,
EventArgs e)
{
    if (this.fatherName_textBox.Text == empty_error_str)
        this.fatherName_textBox.Clear();
}

private void year_textBox_Click(object sender, EventArgs e)
{
    if (this.year_textBox.Text == empty_error_str ||
this.year_textBox.Text == incorrect_num_error_str)
        this.year_textBox.Clear();
}

private void job_textBox_Click(object sender, EventArgs e)
{
    if (this.job_textBox.Text == empty_error_str)
        this.job_textBox.Clear();
}
}
}

```


Код третьей формы:

```
using System;
using System.Windows.Forms;

namespace lab4_windowsApp
{
    public partial class accountForm : Form
    {
        public static Bank_Account curAcc = null;
        public accountForm()
        {
            InitializeComponent();
            int[] accounts =
InputClientForm.curClient.GetAccounts();
            comboBox1.Text = "Выберете счет";
            for (int i = 0; i < accounts.Length; i++)
                comboBox1.Items.Add(accounts[i]);
        }
        private void button1_Click(object sender, EventArgs e)
        {
            Form4 f = new Form4();
            f.ShowDialog();
        }
        private void newAccount_Click(object sender, EventArgs e)
        {
            Bank_Account ba = new Bank_Account();
            startForm.accounts.Add(ba);
            InputClientForm.curClient.addAccount(ba.GetNumber());

            comboBox1.Items.Add(InputClientForm.curClient.GetAccounts()[InputCli
entForm.curClient.GetAccounts().Length - 1]);
            curAcc = ba;
            Form4 f = new Form4();
            f.ShowDialog();
        }
    }
}
```

Код четвертой формы:

```
using System;
using System.Windows.Forms;

namespace lab4_windowsApp
{
    public partial class Form4 : Form
    {
        public static string closedAccError = "### BANK ACCOUNT IS
CLOSED ###";
        public Form4()
        {
            InitializeComponent();
            label1.Text = "Balance: ";
            comboBox1.Text = "History: ";
        }
    }
}
```

```

private void openAcc_Click(object sender, EventArgs e)
{
    accountForm.curAcc.OpenAccount();
}
private void button1_Click(object sender, EventArgs e)
{
    accountForm.curAcc.CloseAccount();
}
private void historyView_Click(object sender, EventArgs e)
{
    for (int i = 0; i <
accountForm.curAcc.getHistory().Count; i++)
    {
        comboBox1.Items.Add(accountForm.curAcc.getHistory()[i]);
    }
}
private void inputMoney_Click(object sender, EventArgs e)
{
    double sum;
    if (textBox1.Text == "")
    {
        this.textBox1.Text =
InputClientForm.empty_error_str;
    }
    else
    {
        if (!(double.TryParse(textBox1.Text, out sum)))
        {
            this.textBox1.Text =
InputClientForm.incorrect_num_error_str;
        }
        else
        {
            if (accountForm.curAcc.GetState())
            {
                accountForm.curAcc.in_money(sum);
                this.textBox1.Text = $"### TRANSACTION IN:
{sum} ###";
            }
            else
            {
                this.textBox1.Text = closedAccError;
            }
        }
    }
}
private void OutMoney_Click(object sender, EventArgs e)
{
    double sum;
    if (textBox2.Text == "")
    {
        this.textBox2.Text =
InputClientForm.empty_error_str;
    }
}

```

```

    }
    else
    {
        if (!(double.TryParse(textBox2.Text, out sum)))
        {
            this.textBox2.Text =
InputClientForm.incorrect_num_error_str;
        }
        else
        {
            if (accountForm.curAcc.GetState())
            {
                accountForm.curAcc.out_money(sum);
                this.textBox2.Text = $"### TRANSACTION OUT:
{sum} ###";
            }
            else
            {
                this.textBox2.Text = closedAccError;
            }
        }
    }
}

private void balanceView_Click(object sender, EventArgs e)
{
    label11.Text = "Balance: " +
accountForm.curAcc.getBalance().ToString();
}
}
}

```

Вывод: в процессе выполнения лабораторной работы были изучены различные элементы оконного интерфейса: кнопка, надпись, окно для ввода, выпадающий список. В качестве основной идеи программы был взят материал предыдущей лабораторной работы. Для поставленной задачи был создан четырехоконный интерфейс, позволяющий создавать клиентов и счета, а также работать с ними.