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 09.03.01 **Прикладная информатика**

О Т Ч Е Т

по лабораторной работе № 3

Дисциплина: Разработка приложений на языке C#

Преподаватель _____ _____
(Подпись, дата) (И.О. Фамилия)

Студент гр. ИУ6-72Б _____ И.С. Марчук
(Подпись, дата) (И.О. Фамилия)

Москва, 2022

Цель работы: в этой лабораторной работе обучающиеся закрепляют знания по C#, создавая консольное приложение. Программа представляет собой автоматизированную систему учета банковских сведений.

Задание:

Создать консольное приложение.

Программа представляет собой автоматизированную систему учета банковских сведений.

На каждого клиента банка хранятся следующие сведения:

− Ф.И.О.;

− Возраст;

− Место работы;

− Номера счетов.

На каждом счете хранится информация о текущем балансе и история прихода, расхода. Для каждого клиента может быть создано неограниченное количество счетов. С каждым счетом можно производить следующие действия: открытие, закрытие, вклад денег, снятие денег, просмотр баланса, просмотр истории.

Вся информация должна хранится в массивах. Рекомендуется объекты клиента и счета реализовать в виде классов. Баланс счета организовать в виде свойства только для чтения.

Код программы:

```
using System;
using System.Linq;
using System.Xml.Linq;

namespace Bank
{

    // классы данных
    class Client
    {
        public string name;
        public string surname;
        public string last_name;
        public int age;
        public string work;
        public int[] number_account;
        public bool first;

        public Client(string name_, string surname_, string
patronymic_, int age_, string work_, int[] number_account_, bool
first_)
        {
            name = name_;
            surname = surname_;
            last_name = patronymic_;
            age = age_;
```

```
            work = work_;
            number_account = number_account_;
            first = first_;
        }
    }

    class Bill
    {
        public int number;
        public string[] history_coming;
        public string[] history_expenses;
        public string condition;
        public bool work;
        public int coming;
        public int expenses;
        private int balance;
        public string[] history;

        public Bill(int number_, string[] history_coming_,
string[] history_expenses_, string condition_, bool work_, int
coming_, int expenses_, int balance_, string[] history_)
        {
            number = number_;
            history_coming = history_coming_;
            history_expenses = history_expenses_;
            condition = condition_;
            work = work_;
            coming = coming_;
            expenses = expenses_;
            balance = balance_;
            history = history_;
        }

        public int get_balance()
        {
            return balance;
        }

        public void get_money(int money)
        {
            balance -= money;
        }

        public void put_money(int money)
        {
             balance += money;
        }
    }


    class Program
    {
        static Bill[] bill = new Bill[100];
```

```
        static Client[] clients = new Client[0];
        static int number_of_chet = 0;
        static void Main(string[] args)
        {
            while (true)
            {
                Console.WriteLine("------------------------------
------------------------------");
                Console.WriteLine("------------------------------
------------------------------");
                Console.WriteLine("------------------------------
------------------------------");
                Console.WriteLine("Созадть    нового    клиента?
Введите 'y' или 'n'");
                string a = Console.ReadLine();
                //int kol = 0;
                if (a == "y" || a == "y")
                {
                    Array.Resize<Client>(ref            clients,
clients.Length + 1);
                    clients[clients.Length - 1] = createClient();
                    Console.WriteLine();
                }
                else
                {

                    Console.WriteLine("Начинаю     работу     с
существующими клиентами:");
                    work_with_client();
                    Console.WriteLine();
                }
            }
        }

        static Client createClient()
        {
            Console.Write("Введите имя: ");
            string name = Console.ReadLine();
            Console.Write("Введите Фамилию: ");
            string surname = Console.ReadLine();
            Console.Write("Введите Отчество: ");
            string patronymic = Console.ReadLine();
            Console.Write("Возраст: ");
            int age;
            while(!int.TryParse(Console.ReadLine(), out age))
            {
                Console.WriteLine("Введен     не     корректный
возраст!");
            }
            Console.Write("Место работы: ");
            string work = Console.ReadLine();
            int[] mas = { 0 };
            bool first = true;
```

```
            return new Client(name, surname, patronymic, age,
work, mas, first);

        }

        // поиск клиента
        static int find_client()
        {
            int clientPos = -1;
            int same_counter = 0;
            Console.Write("Поиск: Введите имя:");
            string name = Console.ReadLine();



            for (int i = 0; i < clients.Length; i++)
                if (clients[i].name.Trim().Equals(name))
                {
                    clientPos = i;
                    same_counter++;
                }

            if (same_counter == 0)
            {
                Console.WriteLine("Поиск:     Пользователь     не
найден");
            }
            else if (same_counter == 1)
            {
                Console.WriteLine("Поиск: Пользователь найден!");
            }
            else if (same_counter > 1)
            {
                Console.WriteLine("Поиск:    Найдено    несколько
пользователей с таким именем.");
                Console.Write("Введите Фамилию: ");
                string surname = Console.ReadLine();

                clientPos = -1;
                same_counter = 0;
                for (int i = 0; i < clients.Length; i++)
                    if     (clients[i].name     ==     name     &
clients[i].surname == surname) {
                        clientPos = i;
                        same_counter++;
                    }

                if (same_counter == 0)
                {
                    Console.WriteLine("Поиск:   Пользователь   не
найден");
                }
                else if (same_counter == 1)
```

```
                {
                    Console.WriteLine("Поиск:       Пользователь
найден!");
                }
                else if (same_counter > 1)
                {
                    Console.WriteLine("Поиск:  Найдено  несколько
пользователей с таким именем и фамилией.");
                    Console.Write("Введите Отчество: ");
                    string patronymic = Console.ReadLine();


                    for (int i = 0; i < clients.Length; i++)
                        if    (clients[i].name    ==    name    &
clients[i].surname   ==   surname   &   clients[i].last_name   ==
patronymic)
                        {
                            clientPos = i;
                            same_counter++;
                        }

                    if (same_counter == 0)
                    {
                        Console.WriteLine("Поиск: Пользователь не
найден");
                    }
                    else if (same_counter == 1)
                    {
                        Console.WriteLine("Поиск:    Пользователь
найден!");
                    }
                    else if (same_counter > 1)
                    {
                        Console.WriteLine("Поиск:         Найдено
несколько пользователей с таким именем и фамилией. ");
                        Console.WriteLine("Поиск:         Показан
последний добавленный!");
                    }

                }
            }
            return clientPos;
        }


        static void work_with_client()
        {
            String input_buffer;


            // поиск клиента
            int clientPos = find_client();
```

```
            if (clientPos != -1)
            {
                // Пользователь найден
                do
                {

                    Console.WriteLine("Создать    новый    счет?
Введите 'y' или 'n'");
                    string a = Console.ReadLine();
                    Console.WriteLine(" ");
                    if (a == "y" || a == "y")
                    {
                        if (clients[clientPos].first == false)
                        {
                            Array.Resize<int>(ref
clients[clientPos].number_account,
clients[clientPos].number_account.Length + 1);
                        }
                        else
                        {
                            clients[clientPos].first = false;
                        }
                        int                number                =
clients[clientPos].number_account.Length - 1;

                        clients[clients.Length                    -
1].number_account[number] = number_of_chet;
                        number_of_chet++;

                        string[] history_coming = { };
                        string[] history_expenses = { };
                        string condition = "открыт";
                        bool work = true;
                        int coming = 0;
                        int expenses = 0;

                        Console.Write("Введите баланс счета");
                        int balance = 0;
                        while  (!int.TryParse(Console.ReadLine(),
out balance) || balance < 0)
                        {
                            Console.WriteLine("Введено         не
корректное число!");
                        }

                        string[] history = { };

                        bill[number]    =    new    Bill(number,
history_coming,   history_expenses,   condition,   work,   coming,
expenses, balance, history);

                        Console.WriteLine("Новый счет создан");
                    }
```

```
                    else
                    {
                        Console.WriteLine("Выберем один из старых
счетов, с которым будем работать");
                        Console.Write("Счета пользователя: ");
                        Console.WriteLine(string.Join(",",
clients[clients.Length - 1].number_account));
                        Console.WriteLine("Счет, с которым будем
работать: ");
                        int numer_work_chet=0;
                        while  (!int.TryParse(Console.ReadLine(),
out numer_work_chet) || numer_work_chet < 0)
                        {
                            Console.WriteLine("Введено         не
корректное число!");
                        }

                        if
(clients[clientPos].number_account.Contains(numer_work_chet)) {
                            Console.WriteLine();
                            Console.WriteLine("Операция,  которая
будет выполняться?");
                            Console.WriteLine("Введите  1,  если
нужно открыть/закрыть счета");
                            Console.WriteLine("Введите  2,  если
нужно просмотреть баланс");
                            Console.WriteLine("Введите  3,  если
нужно просмотреть историю");
                            Console.WriteLine("Введите  4,  если
нужно положить на счет");
                            Console.WriteLine("Введите  5,  если
нужно снять деньги");
                            string b = Console.ReadLine();

                            switch (b)
                            {
                            case "1":
                                Console.WriteLine("Состояние
счета:");

Console.WriteLine(bill[numer_work_chet].condition);
                                if
(bill[numer_work_chet].condition == "открыт")
                                {
                                    Console.WriteLine("Закрыть
счет? y/n");
                                    a = Console.ReadLine();
                                    if (a == "y" || a == "y")
                                    {

bill[numer_work_chet].condition = "закрыт";

bill[numer_work_chet].work = false;
```

```
                                    }
                                }
                                else
                                {
                                    Console.WriteLine("Открыть
счет? y/n");
                                    a = Console.ReadLine();
                                    if (a == "y" || a == "y")
                                    {

bill[numer_work_chet].condition = "открыт";

bill[numer_work_chet].work = true;
                                    }
                                }
                                break;
                            case "2":
                                if (bill[numer_work_chet].work ==
true)
                                {
                                    Console.WriteLine("Балланс: "
+ bill[numer_work_chet].get_balance());
                                }
                                else
                                {
                                    Console.WriteLine("Счет
закрыт");
                                }
                                break;
                            case "3":
                                Console.WriteLine("История работы
со счетом:");
                                Console.WriteLine(",     "     +
string.Join(",", bill[numer_work_chet].history));
                                break;
                            case "4":
                                if (bill[numer_work_chet].work ==
true)
                                {
                                    Console.WriteLine("Сумма,
которую собираетесь положить на счет:");
                                    int          money          =
Int32.Parse(Console.ReadLine());

bill[numer_work_chet].put_money(money);
                                    Array.Resize<string>(ref
bill[numer_work_chet].history_coming,
bill[numer_work_chet].history_coming.Length + 1);

bill[numer_work_chet].history_coming[bill[numer_work_chet].histo
ry_coming.Length - 1] = "Внесено: " + money;
```

```
                                    Array.Resize<string>(ref
bill[numer_work_chet].history,
bill[numer_work_chet].history.Length + 1);

bill[numer_work_chet].history[bill[numer_work_chet].history.Leng
th - 1] = "Внесено: " + money;

                                    Console.WriteLine("Балланс: "
+ bill[numer_work_chet].get_balance());
                                }
                                else
                                {
                                    Console.WriteLine("Счет
закрыт");
                                }
                                break;
                            case "5":
                                if (bill[numer_work_chet].work ==
true)
                                {
                                    Console.WriteLine("Сумма,
которую собираетесь снять со счета:");
                                    int          money          =
Int32.Parse(Console.ReadLine());
                                    if          (money          <
bill[numer_work_chet].get_balance())
                                    {

bill[numer_work_chet].get_money(money);
                                        Array.Resize<string>(ref
bill[numer_work_chet].history_expenses,
bill[numer_work_chet].history_expenses.Length + 1);
                                        Array.Resize<string>(ref
bill[numer_work_chet].history,
bill[numer_work_chet].history.Length + 1);

bill[numer_work_chet].history_expenses[bill[numer_work_chet].his
tory_expenses.Length - 1] = "Снято: " + money;

bill[numer_work_chet].history[bill[numer_work_chet].history.Leng
th - 1] = "Снято: " + money;


Console.WriteLine("Балланс:                    "                    +
bill[numer_work_chet].get_balance());
                                    }
                                    else
                                    {
                                        Console.WriteLine("На
счету недостаточно средств");
                                    }

                                }
```

```
                                    else
                                    {
                                        Console.WriteLine("Счет
закрыт");
                                    }
                                    break;

                                }
                            }
                            else
                                Console.WriteLine("Такого       счета
нет!");
                        }

                        Console.WriteLine("Завершить   редактирование
клиента? Введите y/n");
                        input_buffer = Console.ReadLine();
                    } while (!(String.Compare(input_buffer, "y") == 0
|| String.Compare(input_buffer, "y") == 0));
                }
            }
        }
    }
```

Работа программы показана на рисунках 1 - 3.

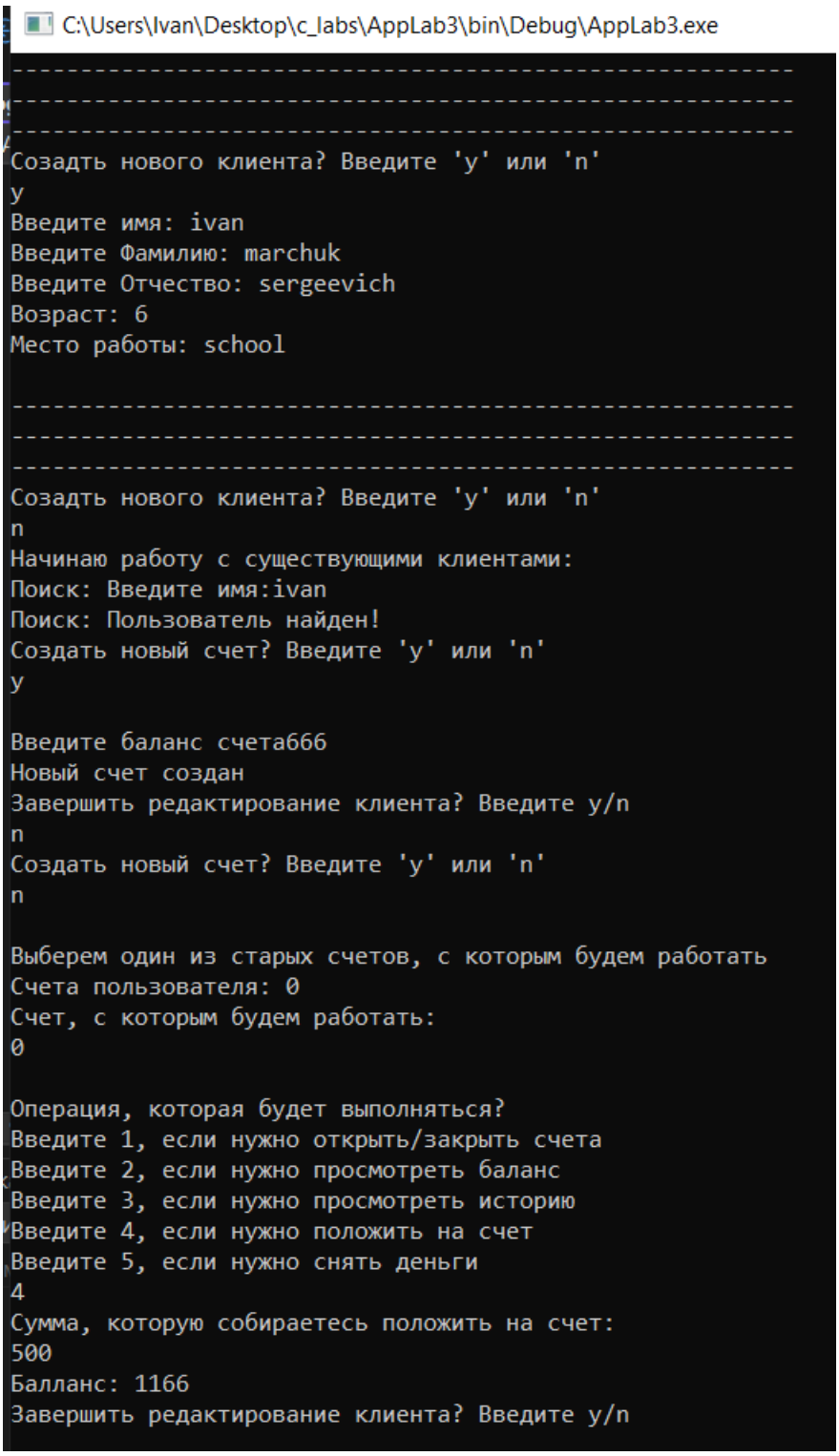

Рисунок 1 – Создание клиента, счета у него и занесение средств на счет

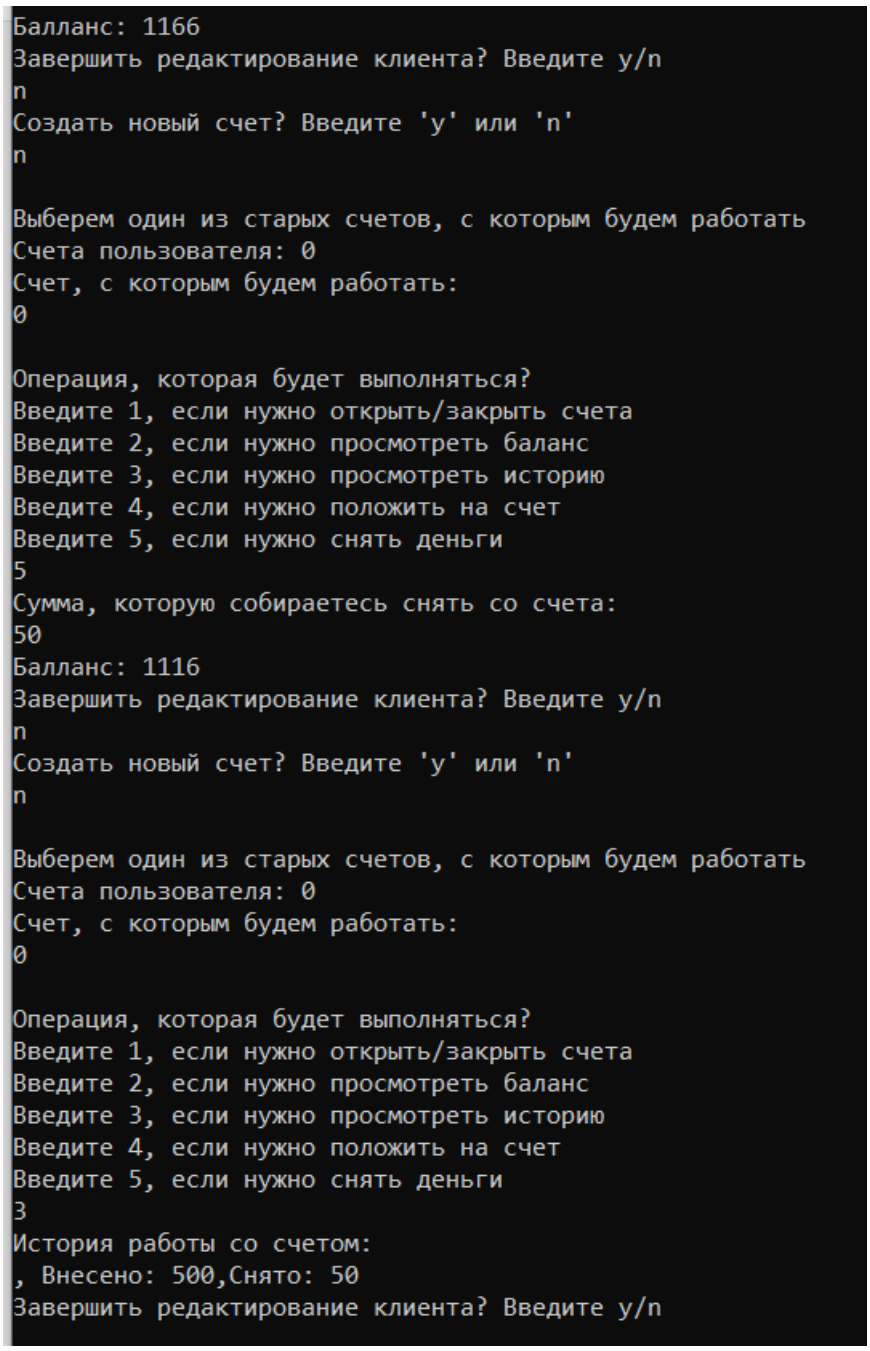

Рисунок 2 – Добвление денег на счет, снятие и просмотр истории операций

Вывод: В ходе лабораторной работы мною была разработана небольшая банковская система, в которой на каждого клиента банка хранятся следующие сведения: Ф.И.О., возраст, место работы, номера счетов, информация о текущем балансе и история прихода, расхода. Также для каждого клиента может быть создано неограниченное количество счетов. С каждым счетом можно производить следующие действия: открытие, закрытие, вклад денег, снятие денег, просмотр баланса, просмотр истории.