Министерство образования Республики Беларусь
Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий
Кафедра технологий программирования

Лабораторная работа №1 по курсу «Объектно-ориентированные технологии программирования и стандарты проектирования»

«Создание классов и объектов»
Вариант 8

Выполнил — Студент гр. 21-ИТ-1
Макеёнок Д.И.

Проверил3 — Ассистент
Хирьянов И.Д.

Полоцк, 2023г.

Цель работы: Научиться создавать классы и объекты этих классов, вызывать методы и обращаться к данным через методы этих объектов, передавать объекты других классов через методы, понимать влияние модификаторов доступа на видимость данных и методов.

Ход работы

Задание:

1. Создать приложение согласно выбранного варианта.
2. В приложении должно быть описано 2 класса (основной и дополнительный), в каждом из которых должны быть закрытые (private) данные и открытые (public) методы.
3. Функции, которые должны быть реализованы в приложении: добавление, удаление, редактирование и просмотр
4. Хотя бы в одном из классов должно быть несколько методов с одинаковыми именами, но с разным набором параметров для инициализации сразу нескольких атрибутов класса.
5. При этом в основном классе должен быть массив объектов дополнительного класса.
6. Для добавления в этот массив должен использоваться метод, в параметрах которому должен передаваться объект дополнительного класса.
7. В теле основной программы необходимо создать объект основного класса, заполнить его данными через методы, заполнить массив объектами дополнительного класса, вывести содержимое основного объекта включая весь массив дополнительных объектов.
8. Сделать выводы.

Вариант 8

Тема проекта: приложение «Отдел кадров».

Спроектировать ПО, предназначенное для создания и просмотра информации по штатным работникам предприятия. Функции, которые должны быть реализованы в приложении: добавление, удаление, редактирование и просмотр работников, структуризация работников по отделам, детальная информация о работнике.

Листинг 1.1 - реализация меню программы:

```
namespace lab1;

class Program
{
    static void Main()
    {
        int dep = 0;
        bool flag;
        Department department = new Department();
        Human Eployment = new Human();
        while (true)
        {
            Console.WriteLine("Что делаем?\n" +
                "1 - Добавить сотрудника\n" +
                "2 - Удалить сотрудника\n" +
                "3 - Просмотреть сотрудников\n" +
                "4 - Редактировать сотрудника\n");
```

```
            int.TryParse(Console.ReadLine(), out int n);
            switch (n)
            {
                case 1:
                    Console.WriteLine("В какой отдел добавить
сотрудника?(1-3)");
                    while (true)
                    {
                        if (int.TryParse(Console.ReadLine(), out dep)
&& dep <= 3 && dep >= 1)
                        {
                            dep--;
                            Eployment.NewEployment();
                            department.AddNewEployee(dep, Eployment);
                            Console.WriteLine("Успешно");
                            // Console.ReadKey();
                            Console.Clear();
                            break;
                        }
                        else Console.Write("Некоректный ввод, попробуй
ещё раз: ");
                    }
                    break;
                case 2:
                    Console.WriteLine("Из какого отдела удалить
сотрудника?(1-3)");
                    while (true)
                    {
                        if (int.TryParse(Console.ReadLine(), out dep)
&& dep <= 3 && dep >= 1)
                        {
                            dep--;
                            flag = department.DeleteEployee(dep);
                            if (flag)
                            {
                                Console.WriteLine("Успешно");
                                break;
                            }
                            else
                            {
                                break;
                            }
                        }
                        else Console.Write("Некоректный ввод, попробуй
ещё раз: ");
                    }
                    Console.ReadKey();
                    Console.Clear();
                    break;
                case 3:
                    Console.WriteLine("Какой отдел вывести?(1-3) / all
- вывести всех");
                    while (true)
                    {
                        string tem = Console.ReadLine();
                        if (int.TryParse(tem, out dep))
                        {
```

```
                            if (dep <= 3 && dep >= 1)
                            {
                                dep--;
                                department.ShowEployee(dep);
                                break;
                            }
                            else Console.WriteLine("Некоректный ввод,
попробуй ещё раз: ");
                        }
                        else
                        {

                            if (tem == "all")
                            {
                                department.ShowEployee();
                                break;
                            }
                            else Console.WriteLine("Некоректный ввод,
попробуй ещё раз: ");
                        }
                    }
                    //Console.ReadKey();
                    //Console.Clear();
                    break;
                case 4:
                    Console.WriteLine("Из какого отдела редактируемый
сотрудник?(1-3)");
                    while (true)
                    {
                        if (int.TryParse(Console.ReadLine(), out dep)
&& dep <= 3 && dep >= 1)
                        {
                            dep--;
                            flag = department.EditingEployee(dep);
                            if (flag)
                            {
                                Console.WriteLine("Успешно");
                                break;
                            }
                            else
                            {
                                break;
                            }

                        }
                        else Console.WriteLine("Некоректный ввод,
попробуй ещё раз: ");
                    }
                    Console.ReadKey();
                    Console.Clear();
                    break;
            }
        }
    }
}
```

Листинг 1.2 - реализация главного класса:

```
using System.Runtime.Intrinsics.Arm;
using System.Security.Cryptography.X509Certificates;

namespace lab1
{
    class Department
    {
        List<Human>[] depart = new List<Human>[3];
        public void AddNewEployee(int num, Human hum)
        {
            depart[num].Add(new Human(hum));
        }
        public Department()
        {
            depart[0] = new List<Human>();
            depart[1] = new List<Human>();
            depart[2] = new List<Human>();
        }

        public bool DeleteEployee(int num)
        {
            int count = 0;
            string name = "";
            bool flag = false, MoreOne = false;

            if (depart[num].Count == 0)
            {
                Console.WriteLine("В этом отделе никого нет.");
                return false;
            }
            Console.WriteLine("Фамилия удаляемого сотрудника:");
            name = Console.ReadLine();
            for (int i = 0; i < depart[num].Count; i++)
            {
                if (depart[num][i].GetSurename().IndexOf(name) != -1)
                {
                    Console.Write(i + ".");
                    depart[num][i].Show();
                    count = i;
                    if (flag)
                    {
                        MoreOne = true;
                    }
                    flag = true;
                }
            }
            if (MoreOne)
            {
                Console.Write("Сотрудника под каким номером нужно
удалить?\n");
                while (true)
                {
                    if (int.TryParse(Console.ReadLine(), out int i))
                    {
                        depart[num].RemoveAt(i);
                        return true;
                    }
```

```
                    else Console.WriteLine("Некоректный ввод, попробуй
ещё раз: ");
                }

            }
            else
            {
                if (flag)
                {
                    depart[num].RemoveAt(count);
                    return true;
                }
                else
                {
                    Console.WriteLine("Неверная фамилия, или такой
нет");
                    return false;
                }
            }

        }
        public void ShowEployee()
        {
            int num = 0;
            for (int i = 0; i < 3; i++)
            {
                Console.WriteLine(++num + "----------------------------
----");
                for (int j = 0; j < depart[i].Count; j++)
                {
                    Console.WriteLine(j + ".");
                    depart[i][j].Show();
                }
                if (depart[i].Count == 0) Console.WriteLine("Тут
пусто\n");
            }
        }
        public void ShowEployee(int num)
        {
            for (int i = 0; i < depart[num].Count; i++)
            {
                Console.WriteLine(i + ".");
                depart[num][i].Show();
            }
            if (depart[num].Count == 0) Console.WriteLine("Тут
пусто\n");
        }
        public bool EditingEployee(int num)
        {
            int count = 0;
            string name = "";
            bool flag = false, MoreOne = false;
            if (depart[num].Count == 0)
            {
                Console.WriteLine("В этом отделе никого нет.");
                return false;
            }
```

```
            Console.Write("Фамилия редактируемого сотрудника:");
            name = Console.ReadLine();
            for (int i = 0; i < depart[num].Count; i++)
            {
                if (depart[num][i].GetSurename() == name)
                {
                    if (depart[num][i].GetSurename().IndexOf(name) !=
-1)
                    {
                        Console.Write(i + ".");
                        depart[num][i].Show();
                        count = i;
                        if (flag)
                        {
                            MoreOne = true;
                        }
                        flag = true;
                    }
                }
            }
            if (MoreOne)
            {
                Console.Write("Сотрудника под каким номером нужно
редактировать?\n");
                while (true)
                {
                    if (int.TryParse(Console.ReadLine(), out int i))
                    {
                        count = i;
                        break;
                    }
                    else Console.WriteLine("Некоректный ввод, попробуй
ещё раз: ");
                }

            }
            else
            {
                if (!flag)
                {
                    Console.WriteLine("Неверная фамилия, или такой
нет");
                    return false;
                }

            }
            Console.WriteLine("Что нужно изменить?\n" +
                "1 - Имя сотрудника\n" +
                "2 - Фамилию сотрудника\n" +
                "3 - День рождения сотрудника\n" +
                "4 - Год начала работы сотрудника\n");
            int edit = Convert.ToInt32(Console.ReadLine());
            switch (edit)
            {
                case 1:
                    depart[num][count].AddHumanName();
                    return true;
```

```
                case 2:
                    depart[num][count].AddHumanSurename();
                    return true;
                case 3:
                    depart[num][count].AddHumanBirthday();
                    return true;
                case 4:
                    depart[num][count].AddHumanStartYear();
                    return true;
            }
            return false;
        }
    }
}
```

Листинг 1.3 – реализация дополнительного класса

```
using System;
using System.Linq.Expressions;
using System.Security.Cryptography;

namespace lab1
{
    class Human
    {
        public Human()
        {
            _startYear = 0;
            _name = "пусто";
            _surename = "пусто";
            _day = 0;
            _month = 0;
            _year = 1980;
        }
        public Human(Human human)
        {
            _startYear = human._startYear;
            _name = human._name;
            _surename = human._surename;
            _day = human._day;
            _month = human._month;
            _year = human._year;
        }
        ~Human()
        {
        }

        private string _name;
        private string _surename;
        private int _day;
        private int _month;
        private int _year;
        private int _startYear;

        public string GetName()
        {
            return _name;
        }
```

```
        public int GetDay()
        {
            return _day;
        }
        public int GetMonth()
        {
            return _month;
        }
        public int GetYear()
        {
            return _year;
        }
        public int GetStartYear()
        {
            return _startYear;
        }
        public string GetSurename()
        {
            return _surename;
        }

        public void AddHumanName()
        {
            Console.Write("Введите имя сотрудника: ");
            bool flag = true;
            while (true)
            {
                string str = Console.ReadLine();
                flag = true;
                foreach (char c in str)
                {
                    if (!((c >= 'a' && c <= 'z') || (c >= 'A' && c <=
'Z') || (c >= 'a' && c <= 'я') || (c >= 'A' && c <= 'Я')))
                    {
                        flag = false;
                        break;
                    }
                }
                if (flag)
                {
                    _name = str;
                    break;
                }
                else
                {
                    Console.Write("Некоректный ввод, попробуй ещё раз:
");
                }
            }
        }
        public void AddHumanSurename()
        {
            Console.Write("Введите фамилию сотрудника: ");
            bool flag = true;
            while (true)
            {
                string str = Console.ReadLine();
```

```
                flag = true;
                foreach (char c in str)
                {
                    if (!((c >= 'a' && c <= 'z') || (c >= 'A' && c <=
'Z') || (c >= 'a' && c <= 'я') || (c >= 'A' && c <= 'Я')))
                    {
                        flag = false;
                        break;
                    }
                }
                if (flag)
                {
                    _surename = str;
                    break;
                }
                else
                {
                    Console.Write("Некоректный ввод, попробуй ещё раз:
");
                }
            }
        }

        public void AddHumanBirthday()
        {
            Console.WriteLine("Ввод Данных дня рождения сотрудника");
            Console.Write("Введите день(1 - 31): ");
            while (true)
            {
                if (int.TryParse(Console.ReadLine(), out int day) &&
day <= 31 && day > 0)
                {
                    _day = day;
                    break;
                }
                else
                {
                    Console.Write("Некоректный ввод, попробуй ещё раз:
");
                }
            }
            Console.Write("Введите месяц(1 - 12): ");
            while (true)
            {
                if (int.TryParse(Console.ReadLine(), out int month) &&
month <= 12 && month > 0)
                {
                    _month = month;
                    break;
                }
                else
                {
                    Console.Write("Некоректный ввод, попробуй ещё раз:
");
                }
            }
            Console.Write("Введите год(1960 - 2005): ");
```

```
            while (true)
            {
                if (int.TryParse(Console.ReadLine(), out int year) &&
year <= 2005 && year >= 1960)
                {
                    _year = year;
                    break;
                }
                else
                {
                    Console.Write("Некоректный ввод, попробуй ещё раз:
");
                }
            }
        }

        public void AddHumanStartYear()
        {
            Console.Write("Введите год начала работы(2000 - 2023): ");
            while (true)
            {
                if (int.TryParse(Console.ReadLine(), out int year) &&
year <= 2023 && year >= 2000 && (year - 18) >= _year)
                {
                    _startYear = year;
                    break;
                }
                else
                {
                    Console.Write("Некоректный ввод, попробуй ещё раз:
");
                }
            }
        }

        public void NewEployment()
        {
            AddHumanName();
            AddHumanSurename();
            AddHumanBirthday();
            AddHumanStartYear();
        }

        public void Show()
        {
            Console.Write("Имя и Фамилия - " + _name + " " + _surename
+ "\n");
            Console.Write("День рождения - " + _day + "." + _month +
"." + _year + "\n");
            Console.Write("Год начала работы - " + _startYear +
"\n\n");
        }

    }

}
```

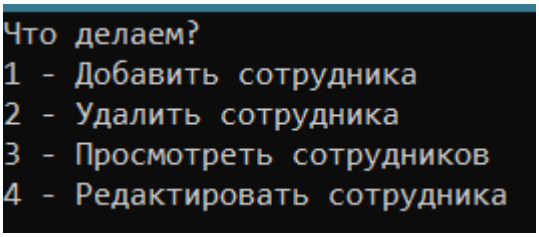

Рисунок 1. Начальное меню

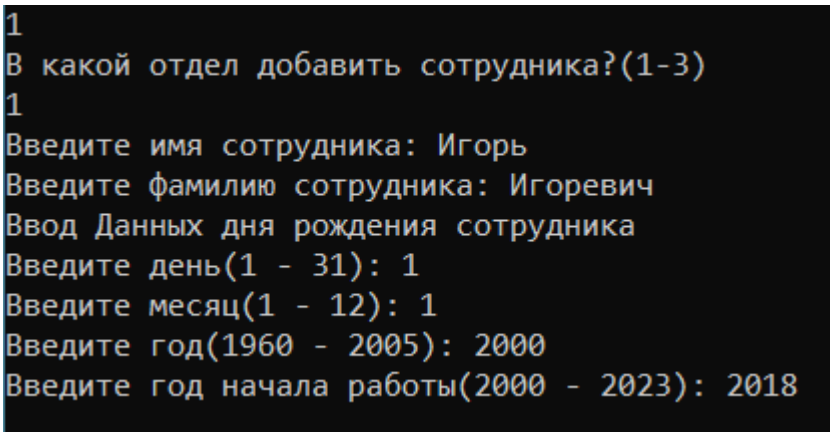

Рисунок 2. Добавление сотрудника

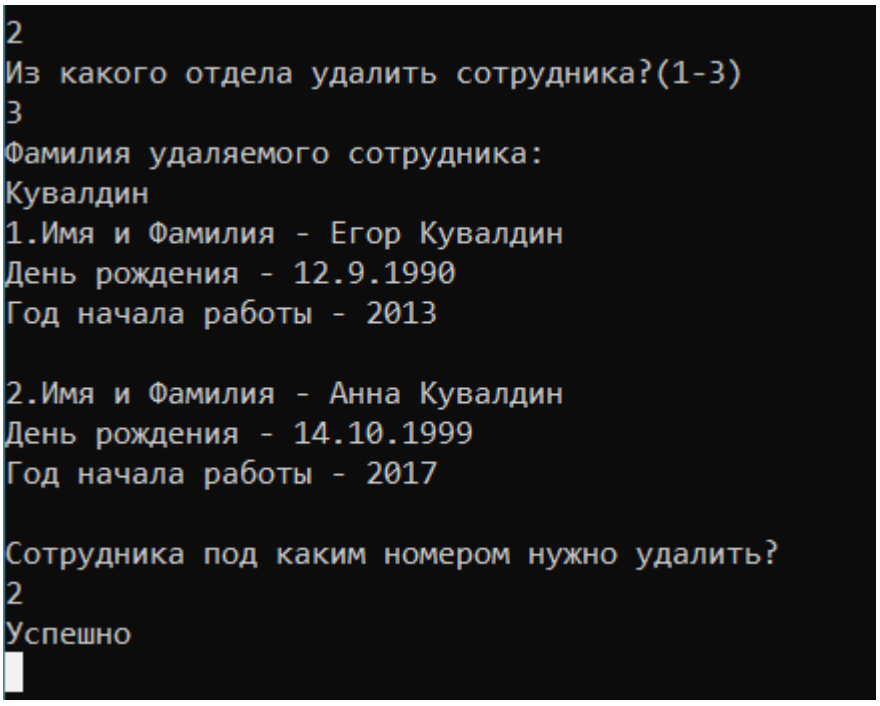

Рисунок 3. Удаление сотрудника

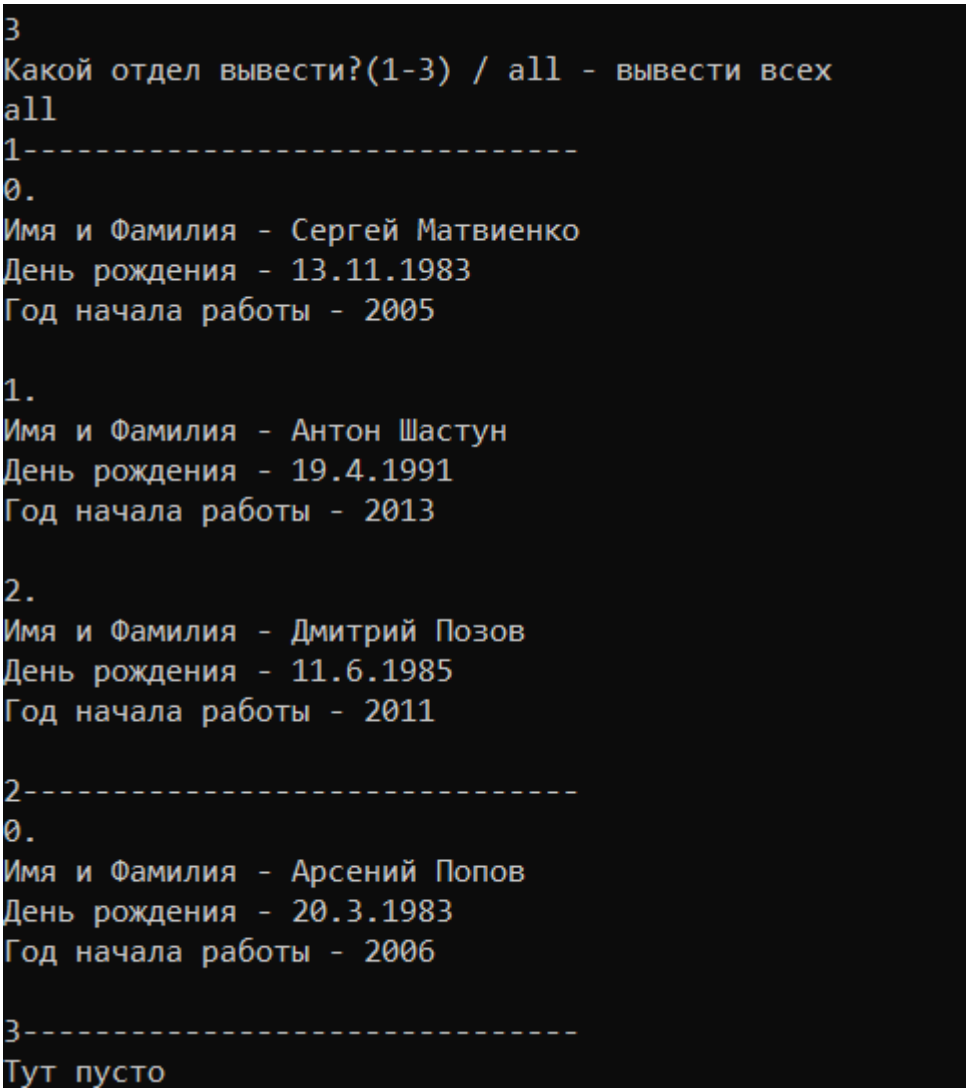

Рисунок 4. Просмотр сотрудников

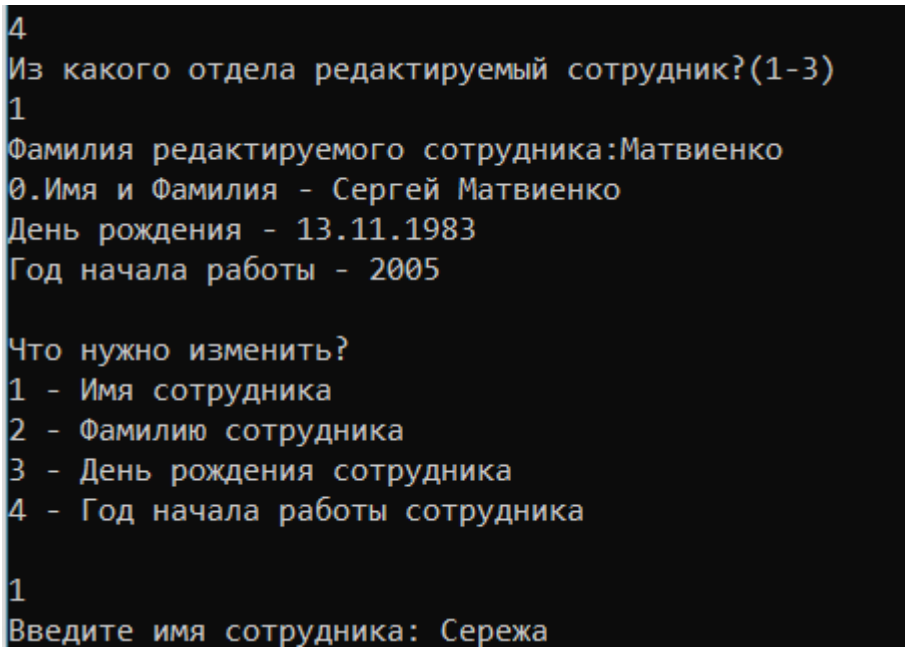

Рисунок 5. Редактирование сотрудника

Вывод: В ходе данный лабораторной работе было изучено как создавать классы и объекты этих классов, вызывать методы и обращаться к данным через методы этих объектов, передавать объекты других классов через методы, понимать влияние

модификаторов доступа на видимость данных и методов. А также применены перезагрузка методов на практике.