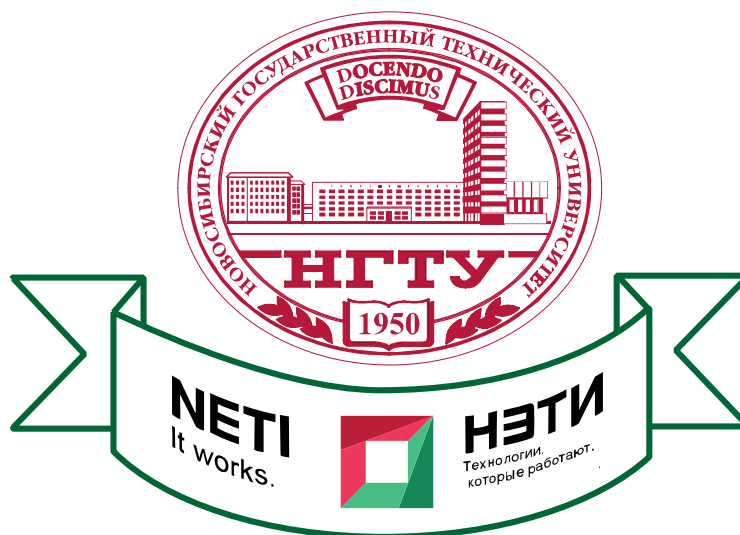


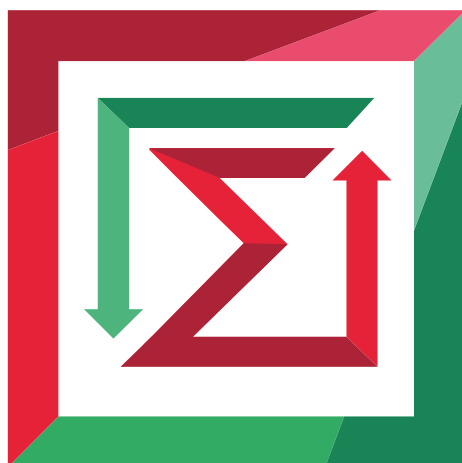
Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Кафедра прикладной математики

Практическое задание № 2
по дисциплине «объектно-ориентированное программирование с
использованием C++/C#»



Факультет:	ПМИ
Группа:	ПМ-71
Студент:	Востриков Вячеслав
Преподаватель:	Ступаков Илья Михайлович

Новосибирск
2019

1. Условие задачи

1. Сделать интерфейс IPerson

```
interface IPerson
{
    string Name { get; }
    string Patronymic { get; }
    string Lastname { get; }
    DateTime Date { get; }
    int Age { get; }
}
```

2. Сделать классы Student и Teacher реализующие этот интерфейс. Они должны содержать конструктор, принимающий значения свойств, статическую функцию создания из строки и переопределение функции ToString.

a. У класса Student дополнительно должны быть свойства: курс, группа и средний балл.

b. У класса Teacher дополнительно должны быть свойства: кафедра, стаж и должность, причем должность должна быть типом enum.

3. Сделать класс University, реализующий свойства и методы из следующего интерфейса.

```
interface IUniversity
{
    IEnumerable<IPerson> Persons { get; } // отсортировать в соответствии с вариантом 1-й лабы
    IEnumerable<Student> Students { get; } // отсортировать в соответствии с вариантом 1-й лабы
    IEnumerable<Teacher> Teachers { get; } // отсортировать в соответствии с вариантом 1-й лабы

    void Add(IPerson person);
    void Remove(IPerson person);

    IEnumerable<IPerson> FindByLastName(string lastName);

    // Для нечетных вариантов. Выдать всех студентов, чей средний балл выше заданного.
    // Отсортировать по среднему баллу
    IEnumerable<Student> FindByAvrPoint(float avrPoint);

    // Для четных вариантов. Выдать всех преподавателей, название кафедры которых содержит
    // заданный текст. Отсортировать по должности.
    IEnumerable<Teacher> FindByDepartment(string text);
}
```

Сделать главную программу с консольным меню, которая бы демонстрировала работу всего вышеперечисленного.

2. Код программы

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;

namespace Task_2
{
    interface IPerson
    {
        string Name { get; }
        string Surname { get; }
        string Lastname { get; }
        DateTime Date { get; }
        int Age { get; }
    }

    class University
    {
        List<IPerson> persons = new List<IPerson>();
        public IEnumerable<Student> Students => persons.OfType<Student>().OrderByDescending(s
=> s.Surname);
        public IEnumerable<Teacher> Teachers => persons.OfType<Teacher>().OrderByDescending(y
=> y.Surname);
        public IEnumerable<IPerson> Persons => persons.OrderByDescending(x => x.Surname);
    }
}
```

```

public IEnumerable<IPerson> Find(string str)
{
    return persons.Where(s => (s.Surname == str));
}

public List<IPerson> Delete(IPerson peop)
{
    persons.Remove(peop);
    return persons;
}

public List<IPerson> Add(IPerson pers)
{
    persons.Add(pers);
    return persons;
}

public IEnumerable<Student> FindSt(float t)
{
    return persons.OfType<Student>().Where(s => s.Avr_score >= t).OrderByDescending(s
=> s.Avr_score);
}

}

public class Student : IPerson
{
    public int Age => Calc.CalcAge(Date);
    public int Course { get; }
    public string Name { get; }
    public string Surname { get; }
    public string Lastname { get; }
    public float Avr_score { get; }
    public string Group { get; }
    public DateTime Date { get; }
    public Student(string surname, string name, string lastname, DateTime date, int
course, string group, float avr_score)
    {
        this.Name = name;
        this.Surname = surname;
        this.Lastname = lastname;
        this.Course = course;
        this.Group = group;
        this.Avr_score = avr_score;
        this.Date = date;
    }

    public static Student Create(string i)
    {
        string[] things = i.Split(new char[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);
        int course = Convert.ToInt32(things[4], 10);
        float avr_score = (float)Convert.ToDouble(things[6]);
        DateTime date = Convert.ToDateTime(things[3]);
        Student student = new Student(things[0], things[1], things[2], date, course,
things[5], avr_score);
        return student;
    }

    override public String ToString()
    {
        return $"student Name: {Surname, 15} {Name, 10} {Lastname, 15}, BirthDate:{Date:
dd.MM.yyyy}, Age:{Age, 4}, Course:{Course, 2}, Group:{Group, 7}, Avr_score:{Avr_score, 5:f2}";
    }
}

```

```

    }

    public class Teacher : IPerson
    {
        public int Age => Calc.CalcAge(Date);
        public float Years { get; set; }
        public string Name { get; }
        public string Surname { get; }
        public string Lastname { get; }
        public enum Dolgnost { Assist, Docent, Professor, St_Prepod, Zav_kaf } //
        public Dolgnost Dolg { get; }
        public string Department { get; }
        public DateTime Date { get; }
        public Teacher(string surname, string name, string lastname, DateTime date, string
department, float years, Dolgnost dolg)
        {
            this.Name = name;
            this.Surname = surname;
            this.Lastname = lastname;
            this.Years = years;
            this.Department = department;
            this.Dolg = dolg;
            this.Date = date;
        }

        public static Teacher Create(string i)
        {
            string[] things = i.Split(new char[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);
            float years = (float)Convert.ToDouble(things[5]);
            Dolgnost dolg = (Dolgnost)Enum.Parse(typeof(Dolgnost), things[6]);
            DateTime date = Convert.ToDateTime(things[3]);
            Teacher teacher = new Teacher(things[0], things[1], things[2], date, things[4],
years, dolg);
            return teacher;
        }

        override public String ToString()
        {
            return $"teacher Name: {Surname,15} {Name,10} {Lastname,15}, BirthDate:{Date:
dd.MM.yyyy}, Age:{Age,4}, Department:{Department,7}, Years:{Years,7:f1}, Dolg:{Dolg,-3}";
        }
    }

    static class Calc
    {
        public static int CalcAge(DateTime Date)
        {
            DateTime dn = DateTime.Now;

            if (dn.Month > Date.Month)
                return (dn.Year - Date.Year);

            else if (dn.Month == Date.Month)
            {
                if (dn.Day < Date.Day)
                    return (dn.Year - Date.Year - 1);
                else
                    return (dn.Year - Date.Year);
            }
            else return (dn.Year - Date.Year - 1);
        }
    }
}

```

```

class Program
{
    public static IPerson ReadPerson(string i)
    {
        string
        flag, k;
        IPerson t;

        flag = i.Substring(0, 1);
        k = i.Substring(2);

        if (flag == "1")
            t = Student.Create(k);
        else
            t = Teacher.Create(k);
        return t;
    }

    public static void write_stud(IEnumerable<Student> students)
    {
        foreach (var a in students)
            Console.WriteLine(a);
    }

    public static void write_teach(IEnumerable<Teacher> teachers)
    {
        foreach (var a in teachers)
            Console.WriteLine(a);
    }

    public static void write_all(IEnumerable<IPerson> persons)
    {
        foreach (var a in persons)
            Console.WriteLine(a);
    }

    static void Main(string[] args)
    {
        string[] text = File.ReadAllLines(@"info.txt");
        University Univer = new University();

        foreach (var b in text)
            Univer.Add(ReadPerson(b));

        Console.WriteLine("Выберите операцию.");
        Console.WriteLine("0 - Выход, 1 - Вывести студентов, 2 - Вывести преподавателей, 3
- Вывести всех, Вывести по среднему баллу, 5 - Поиск по фамилии.");

        int a = Convert.ToInt32(Console.ReadLine());

        while (a != 0)
        {
            switch (a)
            {
                case 1:
                {
                    write_stud(Univer.Students);
                    break;
                }
                case 2:
                {
                    write_teach(Univer.Teachers);

```

```

        break;
    }
    case 3:
    {
        write_all(Univer.Persons);
        break;
    }
    case 4:
    {
        Console.WriteLine("Введите бал: ");
        float t = (float)Convert.ToDouble(Console.ReadLine());
        IEnumerable<Student> B = Univer.FindSt(t);
        write_all(B);
        break;
    }
    case 5:
    {
        Console.WriteLine("Введите фамилию: ");
        string str = Console.ReadLine();
        IEnumerable<IPerson> sup = Univer.Find(str);

        if (sup.Any())
        {
            write_all(sup);
            Console.WriteLine("Удалить из списка? (1 - да, 0 - нет)");
            int k = Convert.ToInt32(Console.ReadLine());

            if (k == 1)
                Univer.Delete(sup.First());

            write_all(Univer.Persons);
        }
        else
            Console.WriteLine("Не найдено.\n");
        break;
    }
    default: { break; }
}
Console.WriteLine("Выберите операцию.");
Console.WriteLine("0 - Выход, 1 - Вывести студентов, 2 - Вывести преподавателей, 3 - Вывести всех, Вывести по среднему баллу, 5 - Поиск по фамилии.");

a = Convert.ToInt32(Console.ReadLine());
    }
}
}

```

3. Результаты работы программы

Исходные данные в info.txt:

```

1 Востриков Вячеслав Альбертович 14.05.1998 3 ПМ-71 4,5
1 Иванов Иван Иванович 21.03.1945 10 ПМ-95 3,5
2 Тракимус Юрий Викторович 01.01.1900 ФПИ 5 Docent

```

Работа программы:

```
Выберите операцию.
0 - Выход, 1 - Вывести студентов, 2 - Вывести преподавателей, 3 - Вывести всех, Вывести по среднему баллу, 5 - Поиск по фамилии.
2
teacher Name:      Тракимус      Юрий      Викторович, BirthDate: 01.01.1900, Age: 119, Department:  ФПМИ, Years:      5,0, Dolg: Docent
Выберите операцию.
0 - Выход, 1 - Вывести студентов, 2 - Вывести преподавателей, 3 - Вывести всех, Вывести по среднему баллу, 5 - Поиск по фамилии.
1
student Name:      Иванов      Иван      Иванович, BirthDate: 21.03.1945, Age: 74, Course:10, Group:  ПМ-95, Avr_score: 3,50
student Name:      Востриков     Вячеслав  Альбертович, BirthDate: 14.05.1998, Age: 21, Course: 3, Group:  ПМ-71, Avr_score: 4,50
Выберите операцию.
0 - Выход, 1 - Вывести студентов, 2 - Вывести преподавателей, 3 - Вывести всех, Вывести по среднему баллу, 5 - Поиск по фамилии.
3
teacher Name:      Тракимус      Юрий      Викторович, BirthDate: 01.01.1900, Age: 119, Department:  ФПМИ, Years:      5,0, Dolg: Docent
student Name:      Иванов      Иван      Иванович, BirthDate: 21.03.1945, Age: 74, Course:10, Group:  ПМ-95, Avr_score: 3,50
student Name:      Востриков     Вячеслав  Альбертович, BirthDate: 14.05.1998, Age: 21, Course: 3, Group:  ПМ-71, Avr_score: 4,50
Выберите операцию.
0 - Выход, 1 - Вывести студентов, 2 - Вывести преподавателей, 3 - Вывести всех, Вывести по среднему баллу, 5 - Поиск по фамилии.
5
Введите фамилию:
Тракимус
teacher Name:      Тракимус      Юрий      Викторович, BirthDate: 01.01.1900, Age: 119, Department:  ФПМИ, Years:      5,0, Dolg: Docent
Удалить из списка? (1 - да, 0 - нет)
1
student Name:      Иванов      Иван      Иванович, BirthDate: 21.03.1945, Age: 74, Course:10, Group:  ПМ-95, Avr_score: 3,50
student Name:      Востриков     Вячеслав  Альбертович, BirthDate: 14.05.1998, Age: 21, Course: 3, Group:  ПМ-71, Avr_score: 4,50
Выберите операцию.
0 - Выход, 1 - Вывести студентов, 2 - Вывести преподавателей, 3 - Вывести всех, Вывести по среднему баллу, 5 - Поиск по фамилии.
1
student Name:      Иванов      Иван      Иванович, BirthDate: 21.03.1945, Age: 74, Course:10, Group:  ПМ-95, Avr_score: 3,50
student Name:      Востриков     Вячеслав  Альбертович, BirthDate: 14.05.1998, Age: 21, Course: 3, Group:  ПМ-71, Avr_score: 4,50
Выберите операцию.
0 - Выход, 1 - Вывести студентов, 2 - Вывести преподавателей, 3 - Вывести всех, Вывести по среднему баллу, 5 - Поиск по фамилии.
2
Выберите операцию.
0 - Выход, 1 - Вывести студентов, 2 - Вывести преподавателей, 3 - Вывести всех, Вывести по среднему баллу, 5 - Поиск по фамилии.
2
Выберите операцию.
0 - Выход, 1 - Вывести студентов, 2 - Вывести преподавателей, 3 - Вывести всех, Вывести по среднему баллу, 5 - Поиск по фамилии.
```