

Министерство образования Республики Беларусь  
Учреждение образования  
«Гомельский государственный университет им. Ф. Скорины»

Отчет по лабораторной работе №6  
**ИНДЕКСАТОРЫ. СТАТИЧЕСКИЕ ПОЛЯ. ПАРАМЕТРИЗОВАННЫЕ  
КЛАССЫ**

Выполнил: студент гр. МС-32 Чвалов К.Р.

Проверил: Шамына А.А.

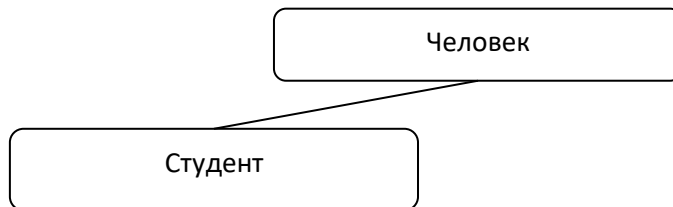
Гомель, 2022

**Цель лабораторной работы:** научиться создавать и использовать интерфейсы.

**Практическая часть:**

**Задание 1.**

Создать иерархию классов:



Класс «человек» должен содержать следующие элементы: поле-фамилия, поле-год рождения, поле статус (**студент, преподаватель, бизнесмен и т.д.**), конструктор с параметрами, конструктор без параметров, метод вывода информации об объекте, **виртуальный** методом **Info** возвращающим возраст. Использовать метод **Info** в методе вывода информации. Реализовать в классе «Человек» интерфейс **Comparable** для сравнения людей по возрасту.

Класс студентов должен содержать дополнительные поля-оценки по математике, физике и истории, конструктор с параметрами, метод для получения среднего балла за сессию. Переопределить метод **Info** в классе «Студент» так, чтобы он возвращал максимальную оценку.

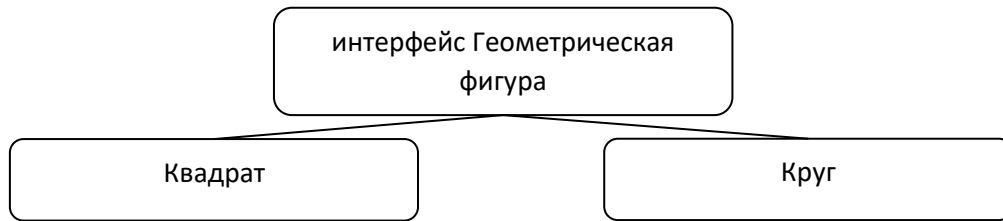
Дополнительно создать класс , реализующий интерфейс **Comparer**. Использовать объект класса для сортировки списка людей по алфавиту.

В методе Main:

- формировать массив объектов класса человек (с клавиатуры не вводить, генератор случайных значений не использовать).
- Сортировать по возрасту.
- Сортировать по фамилии.
- Выводить информацию

## Задание 2.

Создать классы квадратов и окружностей, реализующие общий интерфейс «Геометрические фигуры»:



Интерфейс должен определять следующие элементы: свойство, возвращающее площадь фигуры, метод вывода информации.

Класс квадратов должен содержать следующие элементы: поле-сторону квадрата, конструктор, реализованные элементы интерфейса, метод вычисления периметра. Класс кругов должен содержать следующие элементы: поля - радиус, цвет фигуры, конструктор, реализованные элементы интерфейса.

Создать класс-контейнер для фигур, содержащий поле-массив фигур (ссылку интерфейсного типа), конструктор и реализующий интерфейс IEnumerable.

В классе Program создать объект класса-контейнера с информацией о двух кругах и двух квадратах. Вывести информацию о фигурах, используя для просмотра содержимого контейнера оператор **foreach**.

### Выполнение работы:

Код на C# (<https://github.com/KASSTUSS/UNIVERSITY/tree/main/OOP/LAB6>):

### Задание 1:

```
using System;
using System.Collections.Generic;

namespace lab6
{
    public class Human
    {
        private string lastname;
        private int birth_year;
        private string status;

        public Human(string lastname,
                      int birth_year,
                      string status)
        {
            this.lastname = lastname;
            this.birth_year = birth_year;
            this.status = status;
        }
    }
}
```

```

    }
    public Human() : this("noname", 0, "noname")
    {
        this.lastname = "noname";
        this.birth_year = 0;
        this.status = "noname";
    }
    public string GSlastname
    {
        get { return lastname; }
        set { lastname = value; }
    }
    public int GSbirth_year
    {
        get { return birth_year; }
        set { birth_year = value; }
    }
    public string GSstatus
    {
        get { return status; }
        set { status = value; }
    }
    public virtual void Info()
    {
        DateTime d = DateTime.Now;
        Console.WriteLine("Возраст: " + (Convert.ToInt32(d.Year) -
GSbirth_year));
    }
    public void About()
    {
        Console.WriteLine("");
        Console.WriteLine("Фамилия: " + GSlastname);
        Console.WriteLine("Год рождения: " + GSbirth_year);
        Info();
        Console.WriteLine("Статус: " + GSstatus);
    }
}

public class Student : Human
{
    private int math_mark;
    private int phys_mark;
    private int hist_mark;

    public Student(string lastname,
                    int birth_year,
                    string status,
                    int math_ball,
                    int phys_ball,
                    int hist_ball) : base(lastname, birth_year, status)
    {
        this.math_mark = math_mark;
        this.phys_mark = phys_mark;
        this.hist_mark = hist_mark;
    }
    public int GSmath_mark
    {
        get { return math_mark; }
        set { math_mark = value; }
    }
    public int GSphys_mark
    {
        get { return phys_mark; }
        set { phys_mark = value; }
    }
}

```

```

        public int GShist_mark
        {
            get { return hist_mark; }
            set { hist_mark = value; }
        }
        public float AverageMark()
        {
            float sum_marks = GSmath_mark + GSphys_mark + GShist_mark;
            float average = sum_marks / 3;
            return average;
        }
        public override void Info()
        {
            int[] marks = new int[] { GSmath_mark, GSphys_mark, GShist_mark };
            int max_mark = 0;
            for (int i = 0; i < 3; i++)
            {
                if (marks[i] > max_mark) max_mark = marks[i];
            }
            Console.WriteLine("Максимальная оценка: " + max_mark);
        }
    }

    class AgeSort : IComparer<Human>
    {
        public int Compare(Human first, Human second)
        {
            if (first.GSbirth_year > second.GSbirth_year)
            {
                return 1;
            }
            else if (first.GSbirth_year < second.GSbirth_year)
            {
                return -1;
            }
            else
            {
                return 0;
            }
        }
    }

    class LnameSort : IComparer<Human>
    {
        public int Compare(Human first, Human second)
        {
            if (Convert.ToInt32(first.GSlastname[0]) >
                Convert.ToInt32(second.GSlastname[0]))
            {
                return 1;
            }
            else if (Convert.ToInt32(first.GSlastname[0]) <
                Convert.ToInt32(second.GSlastname[0]))
            {
                return -1;
            }
            else
            {
                return 0;
            }
        }
    }

    public class Program
    {

```

```

public static void Main()
{
    Human
        person1 = new Human("Умнов", 2002, "Студент"),
        person2 = new Human("Прогуляев", 2001, "Студент"),
        person3 = new Human("Незнаев", 2004, "Студент"),
        person4 = new Human("Тригонометров", 1985, "Преподаватель"),
        person5 = new Human("Чвалов", 2003, "Студент");

    Human[] people = new Human[] { person1, person2, person3, person4,
person5 };
    Array.Sort(people, new AgeSort());

    foreach (Human p in people)
    {
        Console.WriteLine("{0} - {1}", p.GSlastname, p.GSbirth_year);
    }
    Console.WriteLine("");

    Array.Sort(people, new LnameSort());
    foreach (Human p in people)
    {
        Console.WriteLine("{0} - {1}", p.GSlastname, p.GSbirth_year);
    }
    foreach (Human p in people)
    {
        p.About();
    }
}
}
}

```

Вывод:

```

Тригонометров - 1985
Прогуляев - 2001
Умнов - 2002
Чвалов - 2003
Незнаев - 2004

Незнаев - 2004
Прогуляев - 2001
Тригонометров - 1985
Умнов - 2002
Чвалов - 2003

Фамилия: Незнаев
Год рождения: 2004
Возраст: 18
Статус: Студент

Фамилия: Прогуляев
Год рождения: 2001
Возраст: 21
Статус: Студент

Фамилия: Тригонометров
Год рождения: 1985
Возраст: 37
Статус: Преподаватель

Фамилия: Умнов
Год рождения: 2002
Возраст: 20
Статус: Студент

Фамилия: Чвалов
Год рождения: 2003
Возраст: 19
Статус: Студент

```

## Задание 2:

```
using System;
using System.Collections;

namespace lab6
{
    public interface GeometricFigure
    {
        double Area { get; }
        string Info(int num);
    }

    public class ArrOfFigures : IEnumerable
    {
        string[] figures;
        public ArrOfFigures(string[] gf)
        {
            figures = gf;
        }
        IEnumerator IEnumerable.GetEnumerator()
        {
            return figures.GetEnumerator();
        }
    }

    public class Square : GeometricFigure
    {
        private int side;
        public Square(int side)
        {
            this.side = side;
        }
        public int Side
        {
            get { return side; }
        }
        public double Area
        {
            get { return Side * Side; }
        }
        public int Perimetr
        {
            get { return 4 * Side; }
        }
        public string Info(int num)
        {
            string num_figure = "Квадрат " + num + "\n\n";
            string s = "Сторона квадрата: " + Side + "\n";
            string p = "Периметр квадрата: " + Perimetr + "\n";
            string a = "Площадь квадрата: " + Area + "\n";
            return "\n" + num_figure + s + p + a;
        }
    }

    public class Circle : GeometricFigure
    {
        private int radius;
        private string color;
        public Circle(int radius, string color)
        {
            this.radius = radius;
            this.color = color;
        }
        public int Radius
        {
            get { return radius; }
        }
        public string Color
    }
```

```

    {
        get { return color; }
    }
    public double Area
    {
        get { return 3.14 * radius * radius; }
    }
    public string Info(int num)
    {
        string num_figure = "Круг " + num + "\n\n";
        string r = "Радиус круга: " + Radius + "\n";
        string c = "Цвет круга: " + Color + "\n";
        string a = "Площадь круга: " + Area + "\n";
        return "\n" + num_figure + r + c + a;
    }
}

public class Program
{
    public static void Main()
    {
        Square figure1 = new Square(10);
        Square figure2 = new Square(13);
        Circle figure3 = new Circle(15, "blue");
        Circle figure4 = new Circle(3, "white");
        string div = "-----";
        string[] figures = new string[] {figure1.Info(1), div,
                                         figure2.Info(2), div,
                                         figure3.Info(1), div,
                                         figure4.Info(2)};

        ArrOfFigures arr = new ArrOfFigures(figures);
        foreach (string a in arr)
        {
            Console.WriteLine(a);
        }
    }
}

```

Вывод:

```

Квадрат 1
Сторона квадрата: 10
Периметр квадрата: 40
Площадь квадрата: 100
-----
Квадрат 2
Сторона квадрата: 13
Периметр квадрата: 52
Площадь квадрата: 169
-----
Круг 1
Радиус круга: 15
Цвет круга: blue
Площадь круга: 706,5
-----
Круг 2
Радиус круга: 3
Цвет круга: white
Площадь круга: 28,259999999999998

```

Вывод: я научился создавать и использовать интерфейсы.