

Московский государственный технический
университет имени Н. Э. Баумана.

Факультет “Радиотехнический”
Кафедра ИУ5 “Системы обработки информации и управления”

Отчёт к лабораторной работе №2
по курсу «Парадигмы и конструкции языков
программирования»

Выполнил:

Студент группы РТ5-31Б

Цыгичко А.Н.

Подпись и дата:

Проверил:

Преподаватель кафедры ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва, 2024 г.

Постановка задачи.

Постановка задачи.

Лабораторная работа №2

Разработать программу, реализующую работу с классами.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Абстрактный класс «Геометрическая фигура» содержит виртуальный метод для вычисления площади фигуры.
3. Класс «Прямоугольник» наследуется от «Геометрическая фигура». Ширина и высота объявляются как свойства (property). Класс должен содержать конструктор по параметрам «ширина» и «высота».
4. Класс «Квадрат» наследуется от «Прямоугольник». Класс должен содержать конструктор по длине стороны.
5. Класс «Круг» наследуется от «Геометрическая фигура». Радиус объявляется как свойство (property). Класс должен содержать конструктор по параметру «радиус».
6. Для классов «Прямоугольник», «Квадрат», «Круг» переопределить виртуальный метод `Object.ToString()`, который возвращает в виде строки основные параметры фигуры и ее площадь.
7. Разработать интерфейс `IPrint`. Интерфейс содержит метод `Print()`, который не принимает параметров и возвращает `void`. Для классов «Прямоугольник», «Квадрат», «Круг» реализовать наследование от интерфейса `IPrint`. Переопределяемый метод `Print()` выводит на консоль информацию, возвращаемую переопределенным методом `ToString()`.

Текст программы:

Program.cs

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            float width = float.Parse(Console.ReadLine());
            float height = float.Parse(Console.ReadLine());
            Rectangle rectangle = new Rectangle(width, height);
        }
    }
}
```

```

        rectangle.Print();

        float length = float.Parse(Console.ReadLine());
        Square square = new Square(length);
        square.Print();

        float radius = float.Parse(Console.ReadLine());
        Circle circle = new Circle(radius);
        circle.Print();

        Console.WriteLine("-----");

        ArrayList geosh = new ArrayList();
        geosh.Add(rectangle);
        geosh.Add(square);
        geosh.Add(circle);

        geosh.Sort();

        foreach (GeometricShape g in geosh)
        {
            g.Print();
        }

        Console.WriteLine("-----");
        List<GeometricShape> geoshList = new List<GeometricShape>();
        geoshList.Add(rectangle);
        geoshList.Add(square);
        geoshList.Add(circle);
        geoshList.Sort();

        for (int i = 0; i < 3; i++)
        {
            geoshList[i].Print();
        }
        Console.WriteLine("-----");
    }
}

```

Circle.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab2
{
    internal class Circle : GeometricShape
    {
        private float m_Radius;
        public float Radius
        {
            get => m_Radius;
            set => m_Radius = value;
        }
    }
}

```

```

        public Circle(float radius)
        {
            Radius = radius;
            m_Name = "Круг";
            GetArea();
        }

        protected override void GetArea()
        {
            m_Area = (float)Math.PI * m_Radius * m_Radius;
        }

        public override string ToString()
        {
            return m_Name + ":\n Радиус: " + m_Radius + "\n Площадь: " +
m_Area;
        }

        public override void Print()
        {
            Console.WriteLine(ToString());
        }
    }
}

```

GeometricShape.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab2
{
    internal abstract class GeometricShape : IComparable, IPrint
    {
        protected string m_Name;
        protected float m_Area;

        protected virtual void GetArea()
        {
        }

        public float GetA()
        {
            return m_Area;
        }

        protected void SetName(string name)
        {
            m_Name = name;
        }

        public int CompareTo(object? o)
        {
            if(o is GeometricShape gesh) return
GetA().CompareTo(gesh.GetA());
            else throw new ArgumentException("Некорректное значение
параметра");
        }
    }
}

```

```

    }

    public virtual void Print()
    {
        Console.WriteLine(m_Area);
    }
}

```

IPrint.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab2
{
    internal interface IPrint
    {
        void Print();
    }
}

```

Rectangle.cs

```

using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;

namespace Lab2
{
    internal class Rectangle : GeometricShape
    {
        protected float m_Width;
        protected float m_Height;

        public Rectangle(float length)
        {
            Width = length;
            Height = length;
            m_Name = "Прямоугольник";
            GetArea();
        }

        public Rectangle(float width, float height)
        {
            Width = width;
            Height = height;
            m_Name = "Прямоугольник";
            GetArea();
        }

        public float Width

```

```

    {
        get => m_Width;
        set => m_Width = value;
    }

    public float Height
    {
        get => m_Height;
        set => m_Height = value;
    }

    protected override void GetArea()
    {
        m_Area = m_Width * m_Height;
    }

    public override string ToString()
    {
        return m_Name + ":\n Ширина: " + m_Width + "\n Высота: " +
m_Height + "\n Площадь: " + m_Area;
    }

    public override void Print()
    {
        Console.WriteLine(ToString());
    }
}
}

```

Square.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab2
{
    internal class Square : Rectangle
    {
        public Square(float length) : base(length)
        {
            Width = length;
            Height = length;
            m_Name = "Квадрат";
            GetArea();
        }

        public override string ToString()
        {
            return m_Name + ":\n Сторона: " + m_Width + "\n Площадь: " +
m_Area;
        }
    }
}

```

Результат:

12

12

Прямоугольник:

Ширина: 12

Высота: 12

Площадь: 144

13

Квадрат:

Сторона: 13

Площадь: 169

41

Круг:

Радиус: 41

Площадь: 5281,017

Прямоугольник:

Ширина: 12

Высота: 12

Площадь: 144

Квадрат:

Сторона: 13

Площадь: 169

Круг:

Радиус: 41

Площадь: 5281,017

Прямоугольник:

Ширина: 12

Высота: 12

Площадь: 144

Квадрат:

Сторона: 13

Площадь: 169

Круг:

Радиус: 41

Площадь: 5281,017
