**Московский государственный технический** **университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по ЛР№3

Выполнил: Проверил:

студент группы ИУ5-34Б преподаватель каф. ИУ5

Сергеев Максим Гапанюк Ю.Е. Подпись и дата: Подпись и дата:

Москва, 2023 г.

**Задача:**

Разработать программу, реализующую работу с коллекциями.

1. Программа должна быть разработана в виде консольного приложения на языке C#.

2. Создать объекты классов «Прямоугольник», «Квадрат», «Круг».

3. Для реализации возможности сортировки геометрических фигур для класса «Геометрическая фигура» добавить реализацию интерфейса IComparable. Сортировка производится по площади фигуры.

4. Создать коллекцию класса ArrayList. Сохранить объекты в коллекцию. Отсортировать коллекцию. Вывести в цикле содержимое коллекции.

5. Создать коллекцию класса List. Сохранить объекты в коллекцию. Отсортировать коллекцию. Вывести в цикле содержимое коллекции.

**Текст программы:**

using System;

using static System.Math;

using System.Collections;

using Figures;

// task 2

Rectangle rectangle = new Rectangle(2, 3);

Square square = new Square(5);

Circle circle = new Circle(1);

//task 4

Console.WriteLine("task 4");

ArrayList a = new ArrayList();

a.Add(rectangle); a.Add(square); a.Add(circle);

a.Sort();

foreach (var i in a)

Console.WriteLine(i);

//task 5

Console.WriteLine("task 5");

List<Figure> figures = new List<Figure>();

figures.Add(rectangle); figures.Add(square); figures.Add(circle);

figures.Sort();

foreach (Figure i in figures)

Console.WriteLine(i);

namespace Figures

{

abstract class Figure

{

public abstract double Area();

}

internal interface IPrint

{

void Print();

}

class Rectangle : Figure, IPrint, IComparable

{

public double Width { get; set; } = 0;

public double Height { get; set; } = 0;

public Rectangle(double width, double height) { Width = width; Height = height; }

public override double Area() { return Height \* Width; }

public override string ToString()

{

return $"{Width:f2} \* {Height:f2} = {Area():f2}";

}

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

public int CompareTo(object? o)

{

if (o is Rectangle rectangle) return Area().CompareTo(rectangle.Area());

else if (o is Square square) return Area().CompareTo(square.Area());

else if (o is Circle circle) return Area().CompareTo(circle.Area());

else throw new ArgumentException("unable to compare");

}

}

class Square : Rectangle, IPrint, IComparable

{

public Square(double Width) : base(Width, Width) { }

public override double Area() { return Width \* Width; }

public override string ToString()

{

return $"{Width:f2} \* {Width:f2} = {Area():f2}";

}

}

class Circle : Figure, IPrint, IComparable

{

public double Radius { get; set; } = 0;

public Circle(double radius) { Radius = radius; }

public override double Area() { return PI \* Radius \* Radius; }

public override string ToString()

{

return $"pi \* {Radius:f2}^2 = {Area():f2}";

}

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

public int CompareTo(object? o)

{

if (o is Rectangle rectangle) return Area().CompareTo(rectangle.Area());

else if (o is Square square) return Area().CompareTo(square.Area());

else if (o is Circle circle) return Area().CompareTo(circle.Area());

else throw new ArgumentException("unable to compare");

}

}

}

**Результат работы программы:**

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, веб-страница

Автоматически созданное описание