МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курский государственный университет»

Кафедра программного обеспечения и администрирования
информационных систем

Отчёт
по лабораторной работе № 1

“Базовые возможности языка C#”

по дисциплине

“Программирование на C#**”**

Выполнил:

студент группы 413 Мусонда Салиму

Проверил:

ст.пр. кафедры ПОиАИС Ураева Е.Е.

Курск, 2020

Цель: изучить базовые возможности языка C# в части реализации основ объектно-ориентированной методологии программирования.

Задачи:

1) Изучить основные понятия Лекции 1;

2) Выполнить задания (см. подраздел Задание) в соответствии с вариантом.

Задание

I. Определить класс (согласно варианту), в котором:

1) конструктор без параметров;
2) конструктор с параметрами;
3) конструктор копирования;
4) деструктор;
5) статический конструктор;
6) статический метод;
7) статическое поле;
8) свойства (классическое, автоматическое, get only, set only);
9) методы с использованием ref/out параметров;
10) переопределить методы Equals, GetHashCode, ToString.

II. Создайте статический класс, содержащий методы математического преобразования над объектом вашего класса или расчёта определённых параметров (уменьшение, поворот, площадь, периметр и т.п.).

III. Добавьте к созданному классу методы расширения (например: проверки возможности упаковки Вашей геометрической фигуры в коробку размера a, b, c).

IV. Создайте и выведите анонимный тип (по образцу Вашего класса).

V. Продемонстрируйте и объясните работу механизма упаковки и распаковки на примере из Вашего класса.

Вариант 10: название фигуры – Кольцо.

Текст программы

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

namespace nanaka
{
    static class Circle
    {
        private static double radius2, PI=Math.PI;
        static Circle()
        {
            radius2 = 6;
        }
        public static void Area(Ring t)
        {
            double area = 2 * (t.radius * t.radius * t.PI);
            Console.WriteLine("the area of the circle is {0} ", area);
        }
// проверки возможности упаковки Вашей геометрической фигуры в коробку
//размера
        public static void proverka(Ring t)
        {
            if (radius2 - 0.2 >= t.radius)
                Console.WriteLine("circle can fit in the box ");
            else
                Console.WriteLine(" circle can not fit in the box ");

        }


    }
    public class Ring
    {
        // Fields

        public float PI { get; set; }  //Static Class Fields
        static double pi = Math.PI;
        public double radius { get; set; } //automatic
        //public double diameter { get;}
        private int N { get; set; }
        public double InnerD { get; set; }
        public double OutD { get; set; }
        public double Area
        {
            get { return getArea(); }
        }
        //constructor with parameters
        public Ring(double radius)
        {
            this.radius = radius;
        }
        //Constructor with parametres
        public Ring()
        {
            radius = 0.0;
        }
        //copy constructor
        public Ring(Ring copy)
```

```
{
    radius = copy.radius;
}
//Destructor
~Ring()        {
    Console.WriteLine("Destructor was called");
}
// static constructor
static Ring()
{
    Console.WriteLine("My Static constructor");
    Ring.pi = 3.141F;
}
//Static method
public static double getArea(double r)
{
    //return getArea();
    return pi * r * r;
}
//method
private double getArea()
{
    if (OutD - InnerD < Double.Epsilon)
    {
        throw new Exception("Внешний диаметр кольца больше внутреннего!");
    }
    return (Math.PI * OutD * OutD / 4.0) - (Math.PI * InnerD * InnerD / 4.0);
}

//Ref
static void GetTheRadius(ref double rad)
{
    do
    {
        Console.Write("Radius: ");
        rad = double.Parse(Console.ReadLine());
        if (rad < 0)
            Console.WriteLine("Please enter a positive number");
    } while (rad < 0);
}
//Out
static void GetTheHeight(out double d)
{
    do
    {
        Console.Write("Diameter: ");
        d = double.Parse(Console.ReadLine());
        if (d < 0)
            Console.WriteLine("Please enter a positive number");
    } while (d < 0);
}
public void boxunbox()
{
    int rad = 12;
    // Boxing copies the value of i into object o.
    object o = rad;
    // Change the value of i.
    rad = 24;
    // The change in i doesn't affect the value stored in o.
    System.Console.WriteLine("The value-type value = {0}", rad);
    System.Console.WriteLine("The object-type value = {0}", o);
}

//overide
public override bool Equals(Object obj)
```

```
        {
            if (obj == null || !(obj is Ring))
                return false;
            else
                return N == ((Ring)obj).N;
        }

        public override int GetHashCode()
        {
            return N;
        }

        public override string ToString()
        {
            return N.ToString();
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Ring mafi = new Ring();
            object chiring;
            chiring = (object)Convert.ChangeType(mafi, typeof(object));
            mafi = (Ring)Convert.ChangeType(chiring, typeof(Ring));
            var anony_object = new
            {
                rad = 2,
                pi = 3.14,
                            };
            Console.WriteLine("circle radius: " + anony_object.rad);
            Console.WriteLine("circle pi: " + anony_object.pi);
            //static class output
            Console.WriteLine("Static class output /n");
            //Circle.Area();
            //Cyli.PrintCircumference(2);
            //Ring c3 = Ring.bigger(cc1,c2);
            var ringing = new Ring(3.5);
            Console.WriteLine(ringing.radius); //output: 1
        }
    }
}
```

Тестирование

Тестирование задачи 1 представлено на рисунке 1.

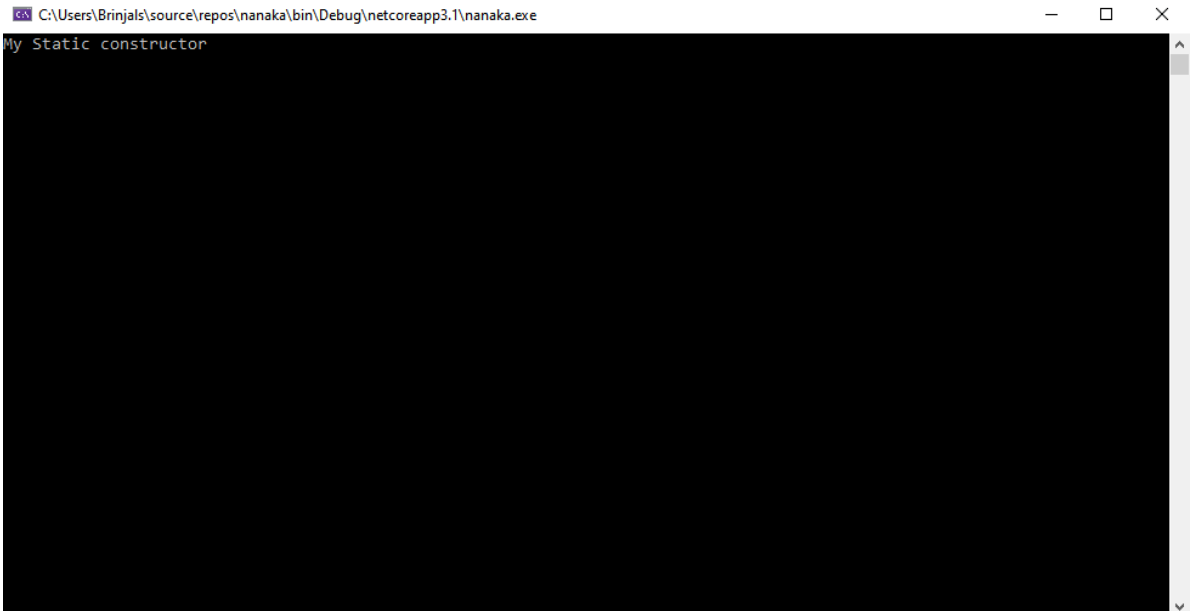

Рисунок 1 – Тестирование задачи 1