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

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №2
«Разработка программы, реализующей работу с классами»

Выполнил:
студент группы ИУ5-34Б
Сергеев Илья

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Подпись и дата:

Подпись и дата:

Москва, 2018 г.

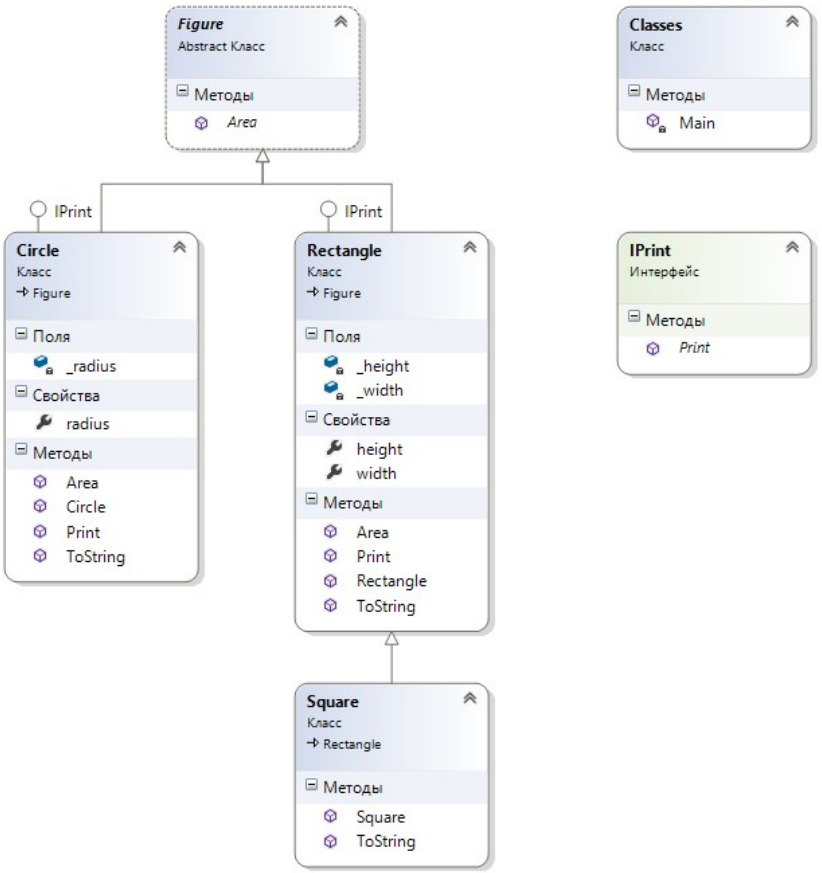
Описание задания

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().

Диаграмма классов

Диаграмма классов генерируется автоматически в среде Visual Studio:



Текст программы (листинг)

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

namespace classes
{
    /// <summary>
    /// Printing interface
    /// </summary>
    interface IPrint
    {
        void Print();
    }

    /// <summary>
    /// Geometric figure class
    /// </summary>
    abstract class Figure
    {
        public abstract double Area();
    }

    /// <summary>
    /// Rectangle class
    /// </summary>
    class Rectangle : Figure, IPrint
    {
        private double _height;
        private double _width;

        public double height
        {
            get { return _height;  }
            set { _height = value; }
        }

        public double width
        {
            get { return _width; }
            set { _width = value; }
        }

        /// <summary>
        /// Constructs a rectangle with width "w" and height "h"
        /// </summary>
        /// <param name="w"></param>
        /// <param name="h"></param>
        public Rectangle(double w, double h)
        {
            height = h;
            width = w;
        }

        /// <summary>
        /// Encalculates an area of the retangle
        /// </summary>
        /// <returns></returns>
        public override double Area()
        {
            return height * width;
        }
```

```
        /// <summary>
        /// Converts an information about this rectangle to "String"
        /// </summary>
        /// <returns></returns>
        public override string ToString()
        {
            return "Rectangle;\nHeight is " + this.height.ToString() + "\n Width is " +
this.width.ToString() + "\n  Area is " + this.Area().ToString();
        }

        /// <summary>
        /// Outputs the an information about the rectangle
        /// </summary>
        public void Print()
        {
            Console.WriteLine(this.ToString());
        }
    }

    /// <summary>
    /// Square class
    /// </summary>
    class Square : Rectangle
    {
        /// <summary>
        /// Constructs a square with side "a"
        /// </summary>
        /// <param name="w"></param>
        /// <param name="h"></param>
        public Square(double a) : base(a, a) { }

        /// <summary>
        /// Converts an information about this square to "String"
        /// </summary>
        /// <returns></returns>
        public override string ToString()
        {
            return "Square;\nSide is " + this.height.ToString() + "\nArea is " +
this.Area().ToString();
        }
    }

    /// <summary>
    /// Circle class
    /// </summary>
    class Circle : Figure, IPrint
    {
        private double _radius;

        public double radius
        {
            get { return _radius; }
            set { _radius = value; }
        }

        /// <summary>
        /// Constructs a circle with radius "r"
        /// </summary>
        /// <param name="w"></param>
        /// <param name="h"></param>
        public Circle(double r) { radius = r; }

        /// <summary>
        /// Encalculates an area of the circle
        /// </summary>
```

```
        /// <returns></returns>
        public override double Area()
        {
            return 2 * System.Math.PI * radius;
        }

        /// <summary>
        /// Converts an information about this circle to "String"
        /// </summary>
        /// <returns></returns>
        public override string ToString()
        {
            return "Circle;\nRadius is " + this.radius.ToString() + "\nArea is " +
this.Area().ToString();
        }

        /// <summary>
        /// Outputs the an information about the circle
        /// </summary>
        public void Print()
        {
            Console.WriteLine(this.ToString());
        }
    }

    class Classes
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Testing functions...\n");

            Rectangle r1 = new Rectangle(3, 4); //creating new rectangle
            r1.Print();
            Console.WriteLine('\n');

            Square s1 = new Square(2); //creating new square
            s1.Print();
            Console.WriteLine('\n');

            Circle c1 = new Circle(7); //creating new circle
            c1.Print();

            Console.ReadKey(); //delay for the user
        }
    }
}
```

Экранные формы с примерами выполнения программы (скриншоты)