**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Дисциплина «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №2

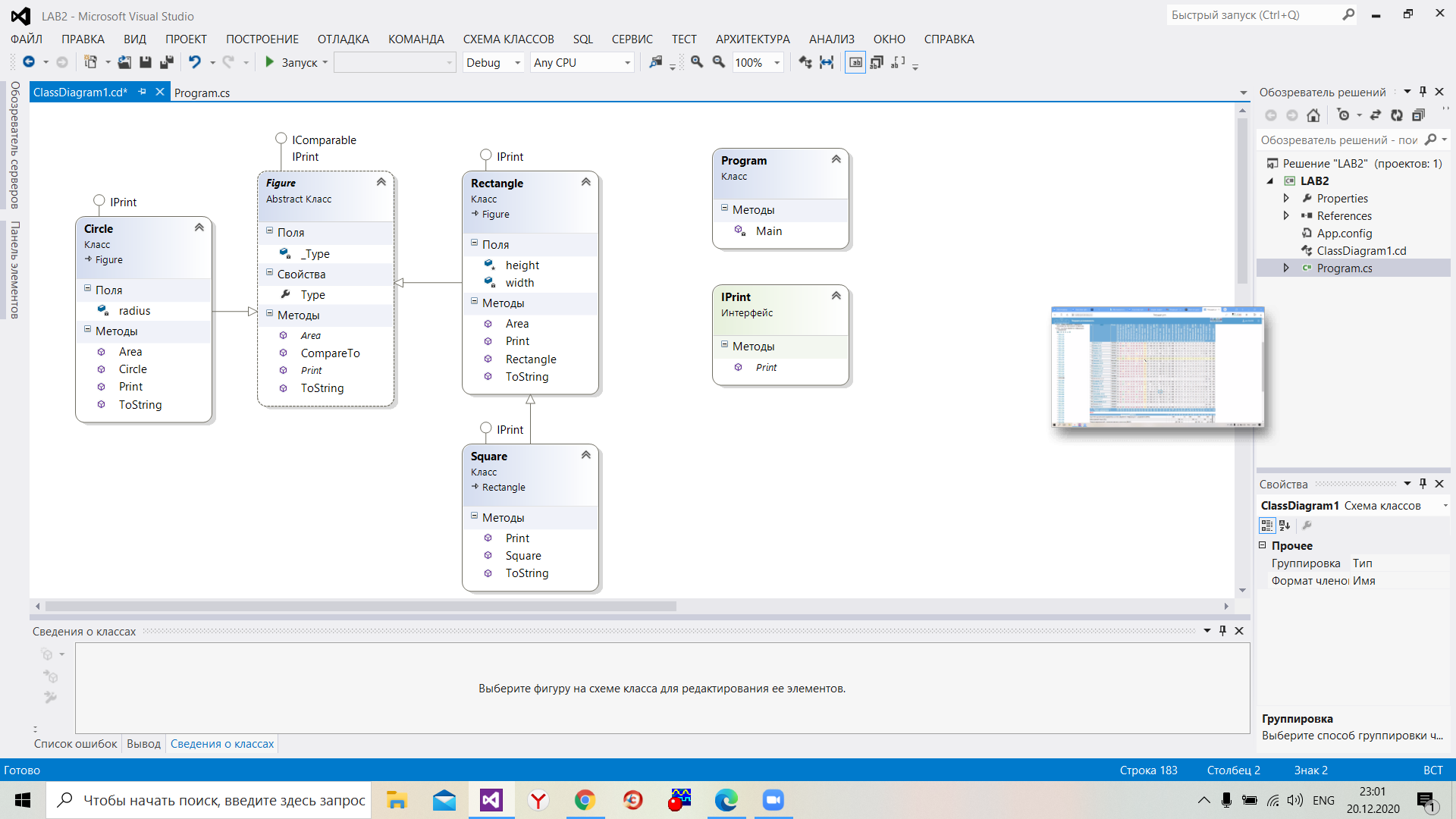
|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы  ИУ5-35Б |  | преподаватель каф. ИУ5 |
| Озеров Павел |  | Гапанюк Ю.Е. |

**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**



**Текст программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FigureCollections

{

/// <summary>

/// Класс фигура

/// </summary>

abstract class Figure : IComparable, IPrint

{

/// <summary>

/// Тип фигуры

/// </summary>

public string Type

{

get

{

return this.\_Type;

}

protected set

{

this.\_Type = value;

}

}

string \_Type;

/// <summary>

/// Вычисление площади

/// </summary>

/// <returns></returns>

public abstract double Area();

/// <summary>

/// Приведение к строке, переопределение метода Object

/// </summary>

/// <returns></returns>

public override string ToString()

{

return this.Type + " площадью " + this.Area().ToString();

}

/// <summary>

/// Вывод объекта на консоль

/// </summary>

/// <returns></returns>

public abstract void Print();

/// <summary>

/// Сравнение элементов (для сортировки списка)

/// </summary>

/// <param name="obj"></param>

/// <returns></returns>

public int CompareTo(object obj)

{

Figure p = (Figure)obj;

if (this.Area() < p.Area())

return -1;

else

if (this.Area() == p.Area())

return 0;

else

return 1; //(this.Area() > p.Area())

}

}

class Rectangle : Figure, IPrint

{

/// <summary>

/// Высота

/// </summary>

protected double height;

/// <summary>

/// Ширина

/// </summary>

double width;

/// <summary>

/// Основной конструктор

/// </summary>

/// <param name="ph">Высота</param>

/// <param name="pw">Ширина</param>

public Rectangle(double ph, double pw)

{

this.height = ph;

this.width = pw;

this.Type = "Прямоугольник";

}

/// <summary>

/// Вычисление площади

/// </summary>

public override double Area()

{

double Result = this.width \* this.height;

return Result;

} //переопределяем для прямоугольника

public override string ToString()

{

return this.Type + " площадью " + this.Area().ToString() + " и сторонами " + this.height + " и " + this.width;

} //переопределяем

public override void Print() //переопределяем

{

Console.WriteLine(this.ToString());

}

}

class Square : Rectangle, IPrint

{

public Square(double size)

: base(size, size)

{

this.Type = "Квадрат";

}

public override string ToString() //переопределяем для квадрата

{

return this.Type + " площадью " + this.Area().ToString() + " и стороной " + this.height;

}

public override void Print() //переопределяем для квадрата

{

Console.WriteLine(this.ToString());

}

}

class Circle : Figure, IPrint

{

/// <summary>

/// Ширина

/// </summary>

double radius;

/// <summary>

/// Основной конструктор

/// </summary>

/// <param name="ph">Высота</param>

/// <param name="pw">Ширина</param>

public Circle(double pr)

{

this.radius = pr;

this.Type = "Круг";

}

public override double Area() //переопределяем для круга

{

double Result = Math.PI \* this.radius \* this.radius;

return Result;

}

public override string ToString() //переопределяем для круга

{

return this.Type + " площадью " + Math.Round(this.Area(), 3).ToString() + " и радиусом " + this.radius;

}

public override void Print() //переопределяем для круга

{

Console.WriteLine(this.ToString());

}

}

interface IPrint

{

void Print();

}

internal class Program

{

private static void Main()

{

Figure figure1 = new Circle(9);

Figure figure2 = new Rectangle(10, 11);

figure1.Print();

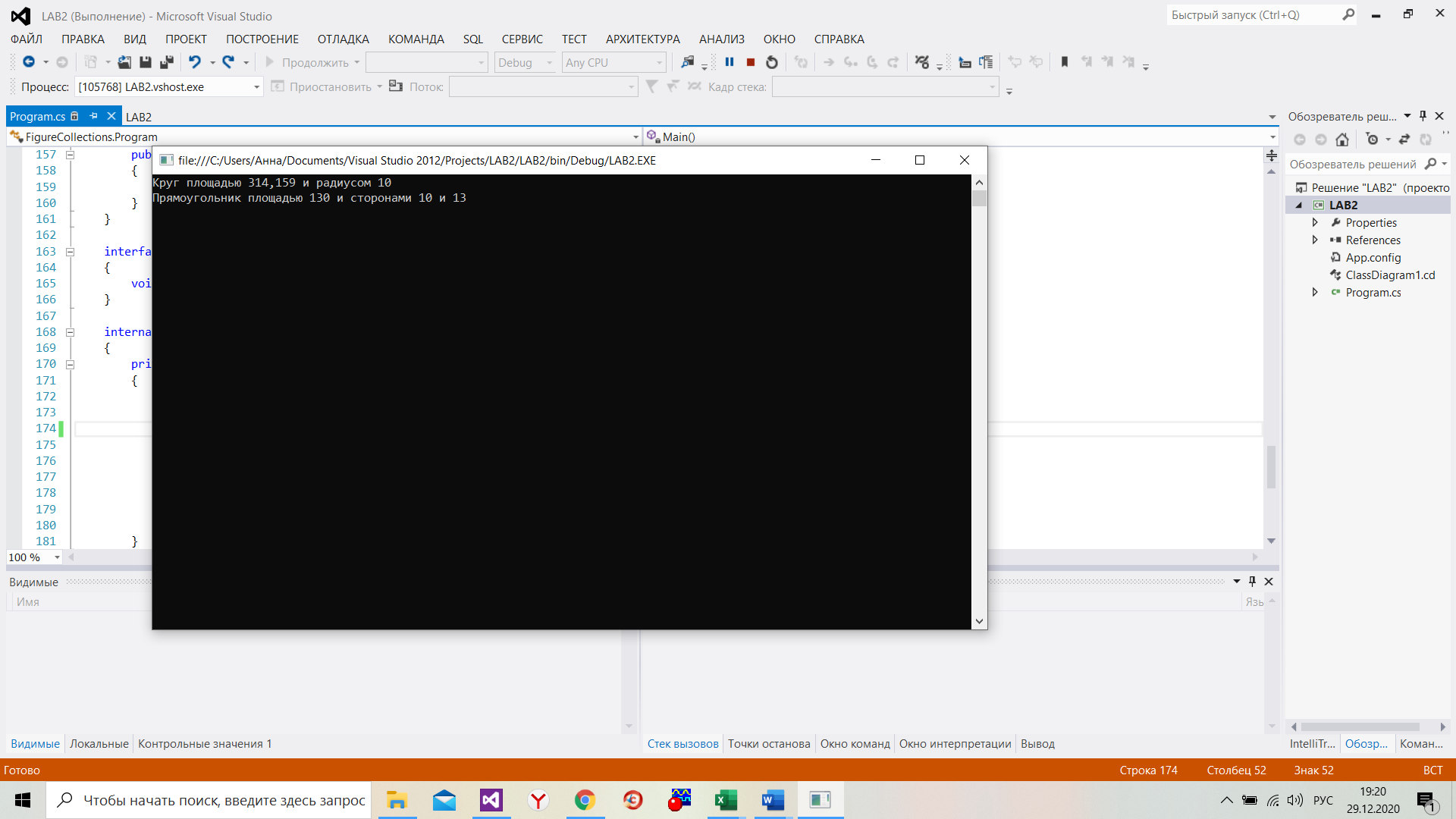
figure2.Print();

Console.ReadLine();

}

}

}

**Результат работы программы:**