**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет технологий»

Отчет по лабораторной работе №2

«Реализация работы с классами»

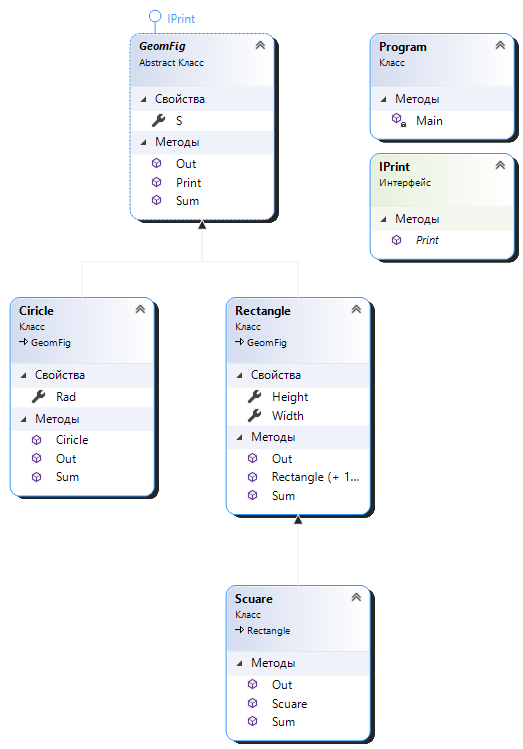
|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Олейников И.И. |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2020 г.

**Условие задания:**

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;

namespace \_2

{

#region Абстрактные

interface IPrint

{

void Print();

}

abstract class GeomFig: IPrint

{

public void Print()

{

Console.WriteLine(Out());

}

public virtual void Sum() { }// нахождение площади

public virtual string Out()//вывод

{

return "\tПлощадь фигуры равна:" + S.ToString();

}

public double S { get; set; }//свойство

}

#endregion

#region Прямоугольник

class Rectangle :GeomFig//прямоугольник

{

public Rectangle() { }//конструктор

public Rectangle(double h, double w)//конструктор

{

Height = h;

Width = w;

}

public double Height { get; set; }//свойство

public double Width { get; set; }//свойство

public override void Sum()

{

S = Height \* Width;

}

public override string Out()//переопр вирт метода возвр значений

{

return "Прямоугольник-\tДлина равна: " + Height.ToString()+ "\tШирина равна: " + Width.ToString() + base.Out();

}

}

#endregion

#region Квадрат

class Scuare : Rectangle //квадрат

{

public override void Sum()

{

S = Height \* Height;

}

public Scuare(double h)//конструктор

{

Height = h;

}

public override string Out()//переопр вирт метода возвр значений

{

return "Квадрат-\tДлина равна: " + Height.ToString()+ "\tПлощадь фигуры равна:" + S.ToString();

}

}

#endregion

#region Круг

class Ciricle : GeomFig//круг

{

public Ciricle(double c)//конструктор

{

Rad = c;

}

public double Rad { get; set; }//свойство

public override void Sum()//переопр вирт метода подсчета площади

{

S = Math.PI \* Rad \* Rad;

}

public override string Out()//переопр вирт метода возвр значений

{

return "Круг-\t\tРадиус равен: " + Rad.ToString()+ base.Out();

}

}

#endregion

class Program

{

static void Main(string[] args)

{

double t1,t2;

#region Ввод даных прямоуольника

Console.WriteLine("Введите первую сторону прямоугольника");

while (!double.TryParse(Console.ReadLine(), out t1))//если ввод- число, записываем

Console.WriteLine("Ввод неверен, попробуйте еще раз");

Console.WriteLine("Введите вторую сторону прямоугольника");

while (!double.TryParse(Console.ReadLine(), out t2))//если ввод- число, записываем

Console.WriteLine("Ввод неверен, попробуйте еще раз");

#endregion

Rectangle rec = new Rectangle(t1, t2);// создание объекта класса прямоугольник

#region Ввод данных квадрата

Console.WriteLine("Введите сторону квадрата");

while (!double.TryParse(Console.ReadLine(), out t1))//если ввод- число, записываем

Console.WriteLine("Ввод неверен, попробуйте еще раз");

#endregion

Scuare scu = new Scuare(t1);// создание объекта класса квадрат

#region Ввод данных круга

Console.WriteLine("Введите радиус круга");

while (!double.TryParse(Console.ReadLine(), out t1))//если ввод- число, записываем

Console.WriteLine("Ввод неверен, попробуйте еще раз");

#endregion

Ciricle cir = new Ciricle(t1);// создание объекта класса круг

rec.Sum();// подсчет площади прямоугольника

scu.Sum();// подсчет площади квадрата

cir.Sum();// подсчет площади круга

rec.Print();//вывод информации

scu.Print();//вывод информации

cir.Print();//вывод информации

}

}

}

**Примеры выполнения программы:**

