Отчёт по лабораторной работе №2 по курсу БКИТ

«Работа с классами»

Подготовил

Ионов С.А.

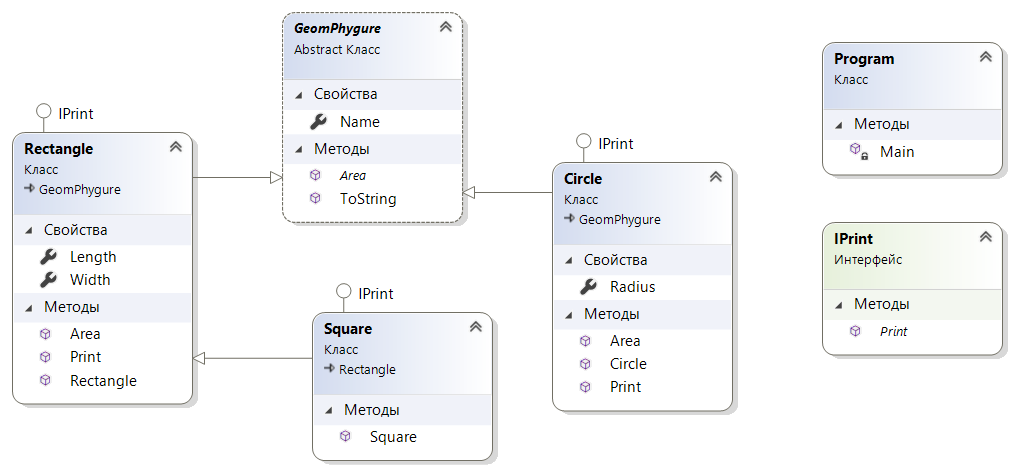
ИУ5-31Б

1. 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().

2. Диаграмма классов



3. Текст программы

using System;

namespace Lab2

{

using Geometry;

class Program

{

static void Main(string[] args)

{

Rectangle rectangle1 = new Rectangle(5, 5);

rectangle1.Print();

Square square1 = new Square(5);

square1.Print();

Circle circle1 = new Circle(5);

circle1.Print();

Console.ReadLine();

}

}

}

namespace Geometry

{

/// <summary>

/// Класс геометрическая фигура

/// </summary>

abstract class GeomPhygure

{

/// <summary>

/// Название фигуры

/// </summary>

public string Name { get; protected set; }

/// <summary>

/// Площадь фигуры

/// </summary>

public abstract double Area();

/// <summary>

/// Переопределение метода ToString(), Краткая информация о фигуре

/// </summary>

public override string ToString()

{

return Name + " имеет площадь " + Area();

}

}

interface IPrint

{

void Print();

}

/// <summary>

/// Класс прямоугольник

/// </summary>

class Rectangle : GeomPhygure, IPrint

{

/// <summary>

/// Длина

/// </summary>

public double Length { get; set; }

/// <summary>

/// Ширина

/// </summary>

public double Width { get; set; }

public Rectangle (double p1, double p2)

{

Length = p1;

Width = p2;

Name = "Прямоугольник";

}

public override double Area()

{

return Length \* Width;

}

public void Print()

{

Console.WriteLine(ToString());

}

}

/// <summary>

/// Класс квадрат

/// </summary>

class Square : Rectangle, IPrint

{

public Square(double size) : base(size, size)

{

Name = "Квадрат";

}

}

/// <summary>

/// Класс круг

/// </summary>

class Circle: GeomPhygure, IPrint

{

/// <summary>

/// Радиус

/// </summary>

public double Radius { get; set; }

public Circle(double p)

{

Radius = p;

Name = "Круг";

}

public override double Area()

{

return Math.PI \* Radius \* Radius;

}

public void Print()

{

Console.WriteLine(ToString());

}

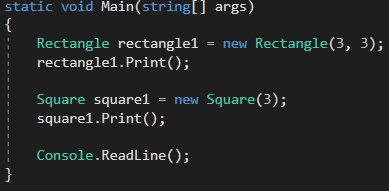
}

}

1. Экранные формы с примерами выполнения программы
2. Программа разбита на классы. В функции Main осуществляется проверка работы классов и выполнение поставленных задач. В данном примере в функции Main были созданы экземпляр класса прямоугольник со сторонами 5 и 6, квадрат со стороной 5 и круг радиусом 5. Далее вызывались методы Print соответствующих классов, которые реализовывали интерфейс IPrint. Информация о результате работы функции Area, вычисляющей площадь фигуры, выводится с помощью переопределённого в абстрактном классе GeomPhygure метода ToString, который возвращает название фигуры и её площадь.



1. Ниже приведён случай, если прямоугольник и квадрат имеют одинаковые стороны. При реализации класса квадрат была использована идея о том, что квадрат – частный случай прямоугольника, то есть не нужно еще раз реализовывать то, что уже написано в данной программе (функция площади Area для прямоугольника).





1. Также можно запрашивать не только площадь фигуры, но также её параметры и устанавливать новые значения для фигуры, не создавая при этом заново экземпляр конкретного класса. Ниже приведён этот случай.

