

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

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №3

«Наследование классов»

Выполнил:

студент группы

ИУ5-32Б

Насруллаев Арсен

Подпись и дата:

Проверил:

преподаватель каф.

ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва, 2023 г.

Задание

Разработать программу, реализующую работу с классами, создать свой класс и выполнить задание по аналогии с фигурами.

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()`.

Код программы

```

using System;

namespace Cars
{
    abstract class Car
    {
        public string Model { get; set; }
        public int Year { get; set; }

        public abstract void Drive();

        public override string ToString()
        {
            return $"Model: {Model}, Year: {Year}";
        }
    }

    class SportsCar : Car, IPrint
    {
        public SportsCar(string model, int year)
        {
            Model = model;
            Year = year;
        }

        public override void Drive()
        {
            Console.WriteLine($"Driving a {Model} sports car!");
        }

        public void Print()
        {
            Console.WriteLine($"Sports Car: {ToString()}");
        }
    }

    class SUV : Car, IPrint
    {
        public SUV(string model, int year)
        {
            Model = model;
            Year = year;
        }

        public override void Drive()
        {
            Console.WriteLine($"Driving a {Model} SUV!");
        }
    }
}

```

```

        public void Print()
        {
            Console.WriteLine($"SUV: {ToString()}");
        }
    }

    class Truck : Car, IPrint
    {
        public Truck(string model, int year)
        {
            Model = model;
            Year = year;
        }

        public override void Drive()
        {
            Console.WriteLine($"Driving a {Model} truck!");
        }

        public void Print()
        {
            Console.WriteLine($"Truck: {ToString()}");
        }
    }

    interface IPrint
    {
        void Print();
    }

    class Program
    {
        static void Main(string[] args)
        {
            SportsCar sportsCar = new SportsCar("Porsche 911", 2021);
            SUV suv = new SUV("Jeep Grand Cherokee", 2022);
            Truck truck = new Truck("Ford F-150", 2020);

            sportsCar.Drive();
            suv.Drive();
            truck.Drive();

            sportsCar.Print();
            suv.Print();
            truck.Print();

            Console.ReadKey();
        }
    }

```

Анализ результатов

```

Driving a Porsche 911 sports car!
Driving a Jeep Grand Cherokee SUV!
Driving a Ford F-150 truck!
Sports Car: Model: Porsche 911, Year: 2021
SUV: Model: Jeep Grand Cherokee, Year: 2022
Truck: Model: Ford F-150, Year: 2020

```

