

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

**Факультет «Радиотехнический»
Кафедра «Системы обработки информации и управления»**

Курс «Парадигмы и конструкции языков программирования»

**Отчет по лабораторной работе №2
«Объектно-ориентированные возможности языка C#»**

Выполнил:
студент группы РТ5-31Б:
Чиканчи В. А.

Подпись и дата:

Проверил:
преподаватель кафедры ИУ5
Гапанюк Ю. Е.

Подпись и дата:

Москва, 2024 г.

Постановка задачи

Разработать программу для решения биквадратного уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Программа осуществляет ввод с клавиатуры коэффициентов A, B, C, вычисляет дискриминант и действительные корни уравнения (в зависимости от дискриминанта).
3. Если коэффициент A, B, C введен некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно.
4. Корни уравнения выводятся зелёным цветом. Если корней нет, то сообщение выводится красным цветом.
5. Коэффициенты A, B, C могут быть заданы в виде параметров командной строки. Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Проверка из пункта 3 в этом случае производится для параметров командной строки без повторного ввода с клавиатуры.

Текст программы

Файл «Program.cs»:

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            float width = float.Parse(Console.ReadLine());
            float height = float.Parse(Console.ReadLine());
            Rectangle rectangle = new Rectangle(width, height);
            rectangle.Print();

            float length = float.Parse(Console.ReadLine());
            Square square = new Square(length);
            square.Print();

            float radius = float.Parse(Console.ReadLine());
            Circle circle = new Circle(radius);
            circle.Print();
        }
    }
}
```

Файл «Circle.cs»:

```

using Lab2;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Xml.Linq;

namespace Lab2
{
    internal class Circle: GeometricShape, IPrint
    {
        private float _radius;

        public float Radius
        {
            get { return _radius; }
            set { _radius = value; }
        }
        public Circle(float radius)
        {
            Radius = radius;
            _name = "Круг";
            GetArea();
        }
        public override void GetArea()
        {
            _area = (float)Math.PI * _radius * _radius;
        }

        public override string ToString()
        {
            return _name + ":\n Радиус: " + _radius + "\n Площадь: " + _area;
        }

        public void Print()
        {
            Console.WriteLine(ToString());
        }
    }
}

```

Файл «GeometricShape.cs»:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab2
{
    internal abstract class GeometricShape
    {
        protected string _name;
        protected float _area;
        public virtual void GetArea()
        {
        }

        public void SetName(string name)
        {
            _name = name;
        }
    }
}

```

```
    }  
}
```

Файл «IPrint.cs»:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace Lab2  
{  
    internal interface IPrint  
    {  
        void Print();  
    }  
}
```

Файл «Rectangle.cs»:

```
using Lab2;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Xml.Linq;  
  
namespace Lab2  
{  
    internal class Rectangle: GeometricShape, IPrint  
    {  
        protected float _width;  
        protected float _height;  
  
        public float Width  
        {  
            get { return _width; }  
            set { _width = value; }  
        }  
        public float Height  
        {  
            get => _height;  
            set => _height = value;  
        }  
        public Rectangle(float length)  
        {  
            Width = length;  
            Height = length;  
            _name = "Прямоугольник";  
            GetArea();  
        }  
        public Rectangle(float width, float height)  
        {  
            Width= width;  
            Height = height;  
            _name = "Прямоугольник";  
            GetArea();  
        }  
        public override void GetArea()  
    }  
}
```

```

    {
        _area = _width * _height;
    }

    public override string ToString()
    {
        return _name + ":\n Ширина: " + _width + "\n Высота: " + _height + "\n
Площадь: " + _area;
    }

    public void Print()
    {
        Console.WriteLine(ToString());
    }
}
}

```

Файл «Program.cs»:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Xml.Linq;

namespace Lab2
{
    internal class Square : Rectangle, IPrint
    {
        public Square(float length) : base(length)
        {
            _name = "Квадрат";
            GetArea();
        }

        public override string ToString()
        {
            return _name + ":\n Сторона: " + _width + "\n Площадь: " + _area;
        }
    }
}

```

Экранные формы с примерами выполнения программы

```

13
15
Прямоугольник:
  Ширина: 13
  Высота: 15
  Площадь: 195
16
Квадрат:
  Сторона: 16
  Площадь: 256
34
Круг:
  Радиус: 34
  Площадь: 3631,6814

```