

Практическая работа №5

Наследование

Разработать классы для объектов, представленных в задании. Необходимо реализовать родительский класс, содержащий абстрактные методы. Реализовать механизм наследования.

В программе необходимо предусмотреть функцию добавления и удаления объекта, а также расчет при выборе объекта из списка.

Объект	Базовый параметр	Доп. параметры	Расчет
Цилиндр	Радиус цилиндра	Высота цилиндра	$V = \pi \cdot x^2 \cdot y$
Куб	Сторона куба		$V = x^3$
Туба	Длина тубы	Внешний радиус Внутренний радиус	$V = \pi \cdot (r_1^2 - r_2^2) \cdot x$

Добавление

Удаление

Редактирование

☒ Цилиндр

☐ Куб

☐ Туба

V=

Цилиндр

X

Y

Куб

X

Туба

X

R1

R2

Цилиндр $x=35$ $y=35$
Куб $x=5$
Туба $r15$ $r2=5$ $x=5$

Добавление
Удаление
Редактирование

☐ Цилиндр
☐ Куб
☒ Туба

V=

Удаление

Добавление
Удаление
Редактирование

☐ Цилиндр
☐ Куб
☒ Туба

V=

Parents.cs

```

abstract class Parents
{
    public double x;

    public Parents(double new_x)
    {
        x = new_x;
    }
    abstract public double S();
}

```

Cl.cs

```

class Cl : Parents
{
    public double y;

    public Cl(double new_x, double new_y) : base(new_x)
    {
        y = new_y;
    }
    public override double S()
    {
        double s = Math.PI * Math.Pow(x, 2) * y;
        return s;
    }
    public override string ToString()
    {
        return "Цилиндр x=" + x + " y="+y;
    }
}

```

Cube.cs

```

class Cube:Parents
{
    public Cube(double new_x) : base(new_x) { }

    public override double S()
    {
        double s = Math.Pow(x, 2);
        return s;
    }
    public override string ToString()
    {
        return "Куб x=" + x;
    }
}

```

Tuba.cs

```

class Tuba:Parents
{
    public double r1;
    public double r2;

    public Tuba(double new_x, double new_r1, double new_r2) : base(new_x)
    {
        r1 = new_r1;
        r2 = new_r1;
    }
    public override double S()
    {
        double s = Math.PI * ((r1 * r1) - (r2 * r2)) * x;
        return s;
    }
    public override string ToString()
    {
        return "Труба r1" + r1 + " r2=" + r2 + " x="+x;
    }
}

```

Form1.cs

```

public partial class Form1 : Form

```

```

{
    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        Form2 f = new Form2();
        if (radioButton1.Checked)
        {
            //Цилиндр
            f.Text = "Добавление цилиндра";
            f.textBox3.Visible = false;
            f.label3.Visible = false;
            f.textBox4.Visible = false;
            f.label4.Visible = false;
            if (f.ShowDialog() == DialogResult.OK)
            {
                double x = Convert.ToDouble(f.textBox1.Text);
                double y = Convert.ToDouble(f.textBox2.Text);
                Cl r = new Cl(x, y);
                listBox1.Items.Add(r);
            }
        }
        else if (radioButton2.Checked)
        {
            f.Text = "Куб";
            f.textBox2.Visible = false;
            f.label2.Visible = false;
            f.textBox3.Visible = false;
            f.label3.Visible = false;
            f.textBox4.Visible = false;
            f.label4.Visible = false;
            if (f.ShowDialog() == DialogResult.OK)
            {
                double x = Convert.ToDouble(f.textBox1.Text);

                Cube cu = new Cube(x);
                listBox1.Items.Add(cu);
            }
        }
        else
        {
            f.Text = "Добавление трубы";
            f.textBox2.Visible = false;
            f.label2.Visible = false;

            if (f.ShowDialog() == DialogResult.OK)
            {
                double r1 = Convert.ToDouble(f.textBox3.Text);
                double r2 = Convert.ToDouble(f.textBox4.Text);
                double x = Convert.ToDouble(f.textBox1.Text);
                Tuba tu = new Tuba(r1, r2, x);
                listBox1.Items.Add(tu);
            }
        }
    }

    private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (listBox1.SelectedIndex != -1)

```

```
        {
            Parents p = listBox1.Items[listBox1.SelectedIndex] as Parents;
            textBox1.Text = p.S().ToString();
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        if (listBox1.SelectedIndex == -1)
        {
            MessageBox.Show("Объект не выбран");
        }
        else
        {
            listBox1.Items.RemoveAt(listBox1.SelectedIndex);
        }
    }
}
```

Ссылка на гитхаб:

<https://github.com/Alexandrov911/PR5.2022.git>