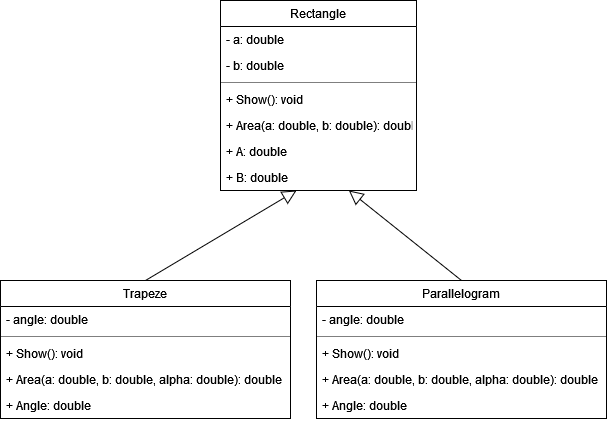
**Лабораторная работа № 4-5**

**Тема:** Определение компонентов ПО Выделение структурных единиц. Разработка спецификаций структурных единиц класса. Организация иерархии наследования. Разработка и отладка программ демонстрации полиморфизма.

**Вариант 8**

**Цель работы:** получить практические навыки разработки иерархии классов, методов, свойств классов.

**Задание:** Создать базовый класс Rectangle (прямоугольник), содержащий поля: длина, ширина. Создать производные классы Parallelogram (параллелограмм) и Trapeze (равнобедренная трапеция), содержащие дополнительное поле: угол наклона. Описать методы для вывода на экран сведений об объекте, получения и установки значений полей, методы вычисления площади фигур. Показать вызов методов на примере массива базового типа, содержащего ссылки на объекты производных типов.



**Класс Rectangle:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace лр4

{

class Rectangle

{

// Поля класса.

private double a;

private double b;

/// <summary>

/// Конструктор класса Rectangle.

/// </summary>

/// <param name="a">Длина прямоугольника.</param>

/// <param name="b">Ширина прямоугольника.</param>

public Rectangle(double a=0.0, double b=0.0)

{

this.a = a;

this.b = b;

}

/// <summary>

/// Свойство доступа к полю a.

/// </summary>

public double A

{

// Установка значения поля a.

set

{

a = value;

}

// Установка значния поля b.

get

{

return a;

}

}

/// <summary>

/// Свойство доступа к полю b.

/// </summary>

public double B

{

// Установка значения поля b.

set

{

b = value;

}

// Получение значения поля b.

get

{

return b;

}

}

/// <summary>

/// Метод вывода объекта класса на консоль.

/// </summary>

public virtual void Show()

{

Console.WriteLine("Сторона a = " + this.a + "\nСторона b = " + this.b);

}

/// <summary>

/// Метод вычисления площади прямоугольника.

/// </summary>

/// <returns>Действительное число, округленное до 3 знака после запятой.</returns>

public virtual double Area()

{

return Math.Round(this.a \* this.b, 3);

}

}

}

**Класс Trapeze:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace лр4

{

class Trapeze : Rectangle

{

//Дополнительное поле класса.

private double angle;

/// <summary>

/// Конструктор класса Trapeze.

/// </summary>

/// <param name="a">Основание трапеции.</param>ыыыы

/// <param name="b">Основание трапеции.</param>

/// <param name="alpha">Угол наклона при основании трапеции.</param>

public Trapeze(double a=0.0, double b=0.0, double alpha=0.0): base(a,b)

{

this.angle = alpha;

}

/// <summary>

/// Метод вывода информации об объекте на консоль.

/// </summary>

public override void Show()

{

base.Show();

Console.WriteLine("Угол наклона = " + angle);

}

/// <summary>

/// Свойство доступа к полю angle.

/// </summary>

public double Angle

{

// Установка значения поля angle.

set

{

angle = value;

}

}

/// <summary>

/// Метод вычисления площади трапеции.

/// </summary>

/// <returns>Дейтсвительное число, округленное до 3-го знака после запятой.</returns>

public override double Area()

{

return Math.Round(((this.B \* this.B - this.A \* this.A) / 2) \* ((Math.Sin(angle \* (Math.PI / 180.0)) \* Math.Sin(angle \* (Math.PI / 180.0))) / Math.Sin(2\*angle \* (Math.PI / 180.0))), 3);

}

}

}

**Класс Parallelogram:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace лр4

{

class Parallelogram : Rectangle

{

// Дополнительное поле класса.

private double angle;

/// <summary>

/// Конструктор класса Parallelogram.

/// </summary>

/// <param name="a">Сторона параллелограмма.</param>

/// <param name="b">Сторона параллелограмма.</param>

/// <param name="alpha">Угол наклона между сторонами параллелограмма.</param>

public Parallelogram(double a=0.0, double b=0.0, double alpha=0.0):base(a, b)

{

this.angle = alpha;

}

/// <summary>

/// Метод вывода информации об объекте класса на консоль.

/// </summary>

public override void Show()

{

base.Show();

Console.WriteLine("Угол наклона = "+this.angle);

}

/// <summary>

/// Свойство доступа к полю angle.

/// </summary>

public double Angle

{

// Установка значения поля angle.

set

{

angle = value;

}

}

/// <summary>

/// Метод вычисления площади параллелограмма.

/// </summary>

/// <returns>Дейтсвительное число, округленное до 3-го знака после запятой.</returns>

public override double Area()

{

return Math.Round(base.Area() \* Math.Sin(angle \* (Math.PI / 180.0)), 3);

}

}

}

**Класс Program:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace лр4

{

class Program

{

static void Main(string[] args)

{

Rectangle[] figures = new Rectangle[6];

figures[0] = new Trapeze(5.5, 7.3, 45);

figures[1] = new Trapeze();

Console.WriteLine("Введите верхнее основание трапеции : ");

figures[1].A = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите нижнее основание трапеции : ");

figures[1].B = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите угол между верхним и нижним основаниями трапеции : ");

((Trapeze)figures[1]).Angle = Convert.ToDouble(Console.ReadLine());

figures[2] = new Parallelogram(3.0, 5.5, 30);

figures[3] = new Rectangle(6.3,10.5);

figures[4] = new Rectangle();

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

figures[4].A = Convert.ToDouble(Console.ReadLine());

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

figures[4].B = Convert.ToDouble(Console.ReadLine());

figures[5] = new Parallelogram();

Console.WriteLine("Введите сторону параллелограмма : ");

figures[5].A = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите вторую сторону параллелограмма : ");

figures[5].B = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите угол между сторонами : ");

((Parallelogram)figures[5]).Angle = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("\nПлощади фигур : \n");

// Цикл вывода массива на консоль.

for (int i = 0; i < figures.Length; i++)

{

string str1 = "";

string str2 = "";

if (figures[i].GetType().ToString()=="лр4.Rectangle")

{

str1 = "Прямоугольник";

str2 = "прямоугольника ";

}

if (figures[i].GetType().ToString() == "лр4.Trapeze")

{

str1 = "Трапеция";

str2 = "трапеции "; ;

}

if (figures[i].GetType().ToString() == "лр4.Parallelogram")

{

str1 = "Параллелограмм";

str2 = "параллелограмма ";

}

Console.WriteLine(str1+" №" + (i + 1));

figures[i].Show();

Console.WriteLine("Площадь "+str2 + figures[i].Area() + "\n");

}

Console.ReadKey();

}

}

}

**Результат выполнения программы:**

Изображение выглядит как текст

Автоматически созданное описание

