**Name: Nguyễn Thị Hà Giang**

**MSV: 22IT072**

src: https://github.com/zannie-hg/Bai-tap-Web-nang-cao/tree/main/baitap\_tuan3/lab\_CS\_OPP1

**LAB C# OPP1**

**A text on a piece of paper

AI-generated content may be incorrect.**

**Circle.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ex1

{

internal class Circle : Shape

{

private int radius;

public Circle(int x, int y, int radius) : base(x, y)

{

this.radius = radius;

}

public override void Show()

{

Console.WriteLine($"Circle with center ({x}, {y}) and radius {radius}");

}

public override string ToString()

{

return $"Circle: Center=({x}, {y}), Radius={radius}";

}

}

}

**Line.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ex1

{

internal class Line : Shape

{

private int x2, y2;

public Line(int x1, int y1, int x2, int y2) : base(x1, y1)

{

this.x2 = x2;

this.y2 = y2;

}

public override void Show()

{

Console.WriteLine($"Line from ({x}, {y}) to ({x2}, {y2})");

}

public override string ToString()

{

return $"Line: ({x}, {y}) -> ({x2}, {y2})";

}

}

}

**PolyLine.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ex1

{

internal class PolyLine : Shape

{

private List<(int, int)> points;

public PolyLine(List<(int, int)> points) : base(points[0].Item1, points[0].Item2)

{

this.points = points;

}

public override void Show()

{

Console.Write("PolyLine with points: ");

foreach (var point in points)

{

Console.Write($"({point.Item1}, {point.Item2}) ");

}

Console.WriteLine();

}

public override string ToString()

{

return $"PolyLine: {string.Join(" -> ", points)}";

}

}

}

**Rectange.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ex1

{

internal class Rectangle : Shape

{

private int x2, y2, x3, y3;

public Rectangle(int x1, int y1, int x2, int y2, int x3, int y3) : base(x1, y1)

{

this.x2 = x2;

this.y2 = y2;

this.x3 = x3;

this.y3 = y3;

}

public override void Show()

{

Console.WriteLine($"Rectangle with points ({x}, {y}), ({x2}, {y2}), ({x3}, {y3})");

}

public override string ToString()

{

return $"Rectangle: ({x}, {y}), ({x2}, {y2}), ({x3}, {y3})";

}

}

}

**Shape.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ex1

{

abstract class Shape

{

protected int x, y;

public Shape(int x, int y)

{

this.x = x;

this.y = y;

}

public void Move(int dx, int dy)

{

x += dx;

y += dy;

Console.WriteLine($"Shape moved to new position: ({x}, {y})");

}

public abstract void Show();

public abstract override string ToString();

}

}

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ex1

{

internal class Program

{

static void Main(string[] args)

{

Shape line = new Line(0, 0, 5, 5);

Shape circle = new Circle(2, 3, 10);

Shape rectangle = new Rectangle(1, 1, 4, 1, 4, 3);

Shape polyLine = new PolyLine(new List<(int, int)> { (1, 1), (2, 3), (4, 5) });

line.Show();

circle.Show();

rectangle.Show();

polyLine.Show();

}

}

}

A screen shot of a computer

AI-generated content may be incorrect.

A white text on a black background

AI-generated content may be incorrect.

**Animal.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ex2

{

abstract class Animal

{

protected string Type;

protected string Name;

public Animal(string type, string name)

{

Type = type;

Name = name;

}

public abstract string MakeSound();

public virtual string GetInfo()

{

return $"Animal Type: {Type}, Name: {Name}";

}

}

}

**Dog.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ex2

{

internal class Dog:Animal

{

private string Breed;

public Dog(string name, string breed) : base("Mammal", name)

{

Breed = breed;

}

public override string MakeSound()

{

return "Woof! Woof!";

}

public override string GetInfo()

{

return base.GetInfo() + $", Breed: {Breed}";

}

}

}

**Cat.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Xml.Linq;

namespace ex2

{

internal class Cat:Animal

{

public Cat(string name) : base("Mammal", name) { }

public override string MakeSound()

{

return "Meow! Meow!";

}

public void Climb(string where)

{

Console.WriteLine($"{Name} is climbing {where}.");

}

}

}

**Duck.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Xml.Linq;

namespace ex2

{

internal class Duck:Animal

{

public Duck(string name) : base("Bird", name) { }

public override string MakeSound()

{

return "Quack! Quack!";

}

public void Swim(string where)

{

Console.WriteLine($"{Name} is swimming in the {where}.");

}

}

}

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ex2

{

internal class Program

{

static void Main(string[] args)

{

Dog dog = new Dog("Buddy", "Spaniel");

Cat cat = new Cat("Whiskers");

Duck duck = new Duck("Daffy");

Console.WriteLine(dog.GetInfo());

Console.WriteLine(dog.MakeSound());

Console.WriteLine(cat.GetInfo());

Console.WriteLine(cat.MakeSound());

cat.Climb("tree");

Console.WriteLine(duck.GetInfo());

Console.WriteLine(duck.MakeSound());

duck.Swim("pond");

}

}

}

A screenshot of a computer

AI-generated content may be incorrect.