**HOMEWORK: Lab3**

**NAME: Đỗ Thành Đạt**

**CLASS: 21se1**

Ex1:

namespace Lab3

{

    public abstract class Shape

    {

        protected double x;

        protected double y;

        public Shape(double x, double y)

        {

            this.x = x;

            this.y = y;

        }

        public void Move(double deltaX, double deltaY)

        {

            x += deltaX;

            y += deltaY;

        }

        public abstract void Show();

        public override string ToString()

        {

            return $"Shape at position: ({x}, {y})";

        }

    }

    public class Line : Shape

    {

        private double x2, y2;

        public Line(double x, double y, double x2, double y2) : base(x, y)

        {

            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: Start ({x}, {y}), End ({x2}, {y2})";

        }

    }

    // Derived class Circle

    public class Circle : Shape

    {

        private double radius;

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

        {

            this.radius = radius;

        }

        public override void Show()

        {

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

        }

        public override string ToString()

        {

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

        }

    }

    // Derived class Rectangle

    public class Rectangle : Shape

    {

        private double x2, y2;

        private double x3, y3;

        public Rectangle(double x, double y, double x2, double y2, double x3, double y3) : base(x, y)

        {

            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: Point 1 ({x}, {y}), Point 2 ({x2}, {y2}), Point 3 ({x3}, {y3})";

        }

    }

    // Derived class PolyLine

    public class PolyLine : Shape

    {

        private List<(double x, double y)> points = new List<(double, double)>();

        public PolyLine(double startX, double startY) : base(startX, startY)

        {

            points.Add((startX, startY));

        }

        public void AddPoint(double x, double y)

        {

            points.Add((x, y));

        }

        public override void Show()

        {

            Console.WriteLine("PolyLine points:");

            foreach (var point in points)

            {

                Console.WriteLine($"({point.x}, {point.y})");

            }

        }

        public override string ToString()

        {

            return $"PolyLine: {points.Count} points";

        }

    }

    class Ex1 {

        public static void run(){

            // Create a Line object

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

            line.Show();

            Console.WriteLine(line);

            line.Move(2, 2);

            line.Show();

            // Create a Circle object

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

            circle.Show();

            Console.WriteLine(circle);

            circle.Move(-1, -1);

            circle.Show();

            // Create a Rectangle object

            Rectangle rect = new Rectangle(0, 0, 4, 0, 0, 6);

            rect.Show();

            Console.WriteLine(rect);

            rect.Move(1, 1);

            rect.Show();

            // Create a PolyLine object

            PolyLine polyline = new PolyLine(0, 0);

            polyline.AddPoint(1, 1);

            polyline.AddPoint(2, 2);

            polyline.Show();

            Console.WriteLine(polyline);

            polyline.Move(1, 1);

            polyline.Show();

        }

    }

}

A screenshot of a computer program

Description automatically generated

Ex2:

namespace Lab3

{

    public abstract class Animal

    {

        public string Type { get; private set; }

        protected Animal(string type)

        {

            Type = type;

        }

        // Abstract method to get sound made by the animal

        public abstract string GetSound();

        // Abstract method to get information about the animal

        public abstract string GetInfo();

    }

    public class Duck : Animal

    {

        public string Name { get; set; }

        public Duck(string name) : base("Bird")

        {

            Name = name;

        }

        public override string GetSound()

        {

            return "Quack";

        }

        public override string GetInfo()

        {

            return $"Duck: Name={Name}";

        }

        public void Swim(string place)

        {

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

        }

        public override string ToString()

        {

            return GetInfo() + $", Sound={GetSound()}";

        }

    }

    public class Dog : Animal

    {

        public string Name { get; set; }

        public string Breed { get; set; }

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

        {

            Name = name;

            Breed = breed;

        }

        public override string GetSound()

        {

            return "Woof";

        }

        public override string GetInfo()

        {

            return $"Dog: Name={Name}, Breed={Breed}";

        }

        public override string ToString()

        {

            return GetInfo() + $", Sound={GetSound()}";

        }

    }

    public class Cat : Animal

    {

        public string Name { get; set; }

        public Cat(string name) : base("Mammal")

        {

            Name = name;

        }

        public override string GetSound()

        {

            return "Meow";

        }

        public override string GetInfo()

        {

            return $"Cat: Name={Name}";

        }

        public void Climb(string place)

        {

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

        }

        public override string ToString()

        {

            return GetInfo() + $", Sound={GetSound()}";

        }

    }

    class Ex2

    {

        public static void run()

        {

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

            Cat cat = new Cat("Kitten");

            Duck duck = new Duck("Ducky");

            Console.WriteLine(dog.ToString());

            Console.WriteLine(cat.ToString());

            Console.WriteLine(duck.ToString());

            cat.Climb("tree");

            duck.Swim("pond");

        }

    }

}

A screen shot of a computer screen

Description automatically generated

Ex3:

namespace Lab3

{

    class Student

    {

        // Các thuộc tính của sinh viên

        public string SID { get; set; }

        public string Name { get; set; }

        public string Department { get; set; }

        public double GPA { get; set; }

        // Constructor để khởi tạo đối tượng Student

        public Student(string sid, string name, string department, double gpa)

        {

            SID = sid;

            Name = name;

            Department = department;

            GPA = gpa;

        }

        // Phương thức hiển thị thông tin sinh viên

        public void DisplayInfo()

        {

            Console.WriteLine($"Mã số sinh viên: {SID}");

            Console.WriteLine($"Tên sinh viên: {Name}");

            Console.WriteLine($"Khoa: {Department}");

            Console.WriteLine($"Điểm TB: {GPA}");

            Console.WriteLine("-------------------------");

        }

    }

    class Ex3

    {

        public static void run()

        {

            Console.Write("Nhập số lượng sinh viên: ");

            int n = int.Parse(Console.ReadLine() ?? "0");

            List<Student> students = new List<Student>();

            for (int i = 0; i < n; i++)

            {

                Console.WriteLine($"Nhập thông tin cho sinh viên thứ {i + 1}:");

                Console.Write("SID: ");

                string sid = Console.ReadLine() ?? "";

                Console.Write("Tên sinh viên: ");

                string name = Console.ReadLine() ?? "Unknow";

                Console.Write("Khoa: ");

                string department = Console.ReadLine() ?? "Unknow";

                Console.Write("Điểm trung bình: ");

                double gpa = double.Parse(Console.ReadLine() ?? "0.0");

                Student student = new Student(sid, name, department, gpa);

                students.Add(student);

            }

            Console.WriteLine("\nDanh sách sinh viên:");

            foreach (Student student in students)

            {

                student.DisplayInfo();

            }

        }

    }

}

A screen shot of a computer

Description automatically generated

Ex4:

namespace Lab3

{

    public class Student

    {

        private string sid;

        private string name;

        private string department;

        private double gpa;

        // Constructor để khởi tạo đối tượng Student

        public Student(string sid, string name, string department, double gpa)

        {

            this.sid = sid;

            this.name = name;

            this.department = department;

            this.gpa = gpa;

        }

        // Getter và Setter cho SID

        public string GetSID() => sid;

        public void SetSID(string value) => sid = value;

        // Getter và Setter cho Name

        public string GetName() => name;

        public void SetName(string value) => name = value;

        // Getter và Setter cho Department

        public string GetDepartment() => department;

        public void SetDepartment(string value) => department = value;

        // Getter và Setter cho GPA

        public double GetGPA() => gpa;

        public void SetGPA(double value) => gpa = value;

        // Phương thức hiển thị thông tin sinh viên

        public void DisplayInfo()

        {

            Console.WriteLine($"SID: {GetSID()}");

            Console.WriteLine($"Tên sinh viên: {GetName()}");

            Console.WriteLine($"Khoa: {GetDepartment()}");

            Console.WriteLine($"Điểm trung bình: {GetGPA():F2}");

            Console.WriteLine("-----------------------------------");

        }

    }

    class Ex4

    {

        public static void run()

        {

            List<Student> students = new List<Student>();

            students = NhapDS();

            XuatDS(students);

        }

        public static List<Student> NhapDS()

        {

            Console.Write("Nhập số lượng sinh viên: ");

            int n = int.Parse(Console.ReadLine() ?? "0");

            List<Student> students = new List<Student>();

            for (int i = 0; i < n; i++)

            {

                Console.WriteLine($"Nhập thông tin cho sinh viên thứ {i + 1}:");

                Student student = Nhap1SV();

                students.Add(student);

            }

            return students;

        }

        public static Student Nhap1SV()

        {

            Console.Write("SID: ");

            string sid = Console.ReadLine();

            Console.Write("Tên sinh viên: ");

            string name = Console.ReadLine();

            Console.Write("Khoa: ");

            string department = Console.ReadLine();

            Console.Write("Điểm trung bình: ");

            double gpa = double.Parse(Console.ReadLine());

            return new Student(sid, name, department, gpa);

        }

        // Hàm xuất danh sách sinh viên

        static void  XuatDS(List<Student> students)

        {

            Console.WriteLine("\nDanh sách sinh viên:");

            foreach (Student student in students)

            {

                student.DisplayInfo();

            }

        }

    }

}

A screenshot of a computer program

Description automatically generated\

Ex5:

using System.Collections.Generic;

using System.Collections;

namespace Lab3

{

    public class People

    {

        private string name;

        private string department;

        // Constructor

        public People(string name, string department)

        {

            this.name = name;

            this.department = department;

        }

        // Getter và Setter cho Name

        public string GetName() => name;

        public void SetName(string value) => name = value;

        // Getter và Setter cho Department

        public string GetDepartment() => department;

        public void SetDepartment(string value) => department = value;

    }

    public class Student : People

    {

        private string sid;

        private double gpa;

        // Constructor

        public Student(string sid, string name, string department, double gpa)

            : base(name, department)

        {

            this.sid = sid;

            this.gpa = gpa;

        }

        // Getter và Setter cho SID

        public string GetSID() => sid;

        public void SetSID(string value) => sid = value;

        // Getter và Setter cho GPA

        public double GetGPA() => gpa;

        public void SetGPA(double value) => gpa = value;

        // Phương thức hiển thị thông tin sinh viên

        public void DisplayInfo()

        {

            Console.WriteLine($"SID: {GetSID()}");

            Console.WriteLine($"Tên sinh viên: {GetName()}");

            Console.WriteLine($"Khoa: {GetDepartment()}");

            Console.WriteLine($"Điểm trung bình: {GetGPA():F2}");

            Console.WriteLine("-----------------------------------");

        }

    }

    class Ex5

    {

        public static void run()

        {

            List<Student> studentList = NhapDS\_List();

            ArrayList studentArrayList = new ArrayList(studentList);

            Console.WriteLine("Danh sách sinh viên từ List<Student>:");

            XuatDS\_List(studentList);

            Console.WriteLine("Danh sách sinh viên từ ArrayList:");

            XuatDS\_ArrayList(studentArrayList);

        }

        static Student Nhap1SV()

        {

            Console.Write("SID: ");

            string sid = Console.ReadLine();

            Console.Write("Tên sinh viên: ");

            string name = Console.ReadLine();

            Console.Write("Khoa: ");

            string department = Console.ReadLine();

            Console.Write("Điểm trung bình: ");

            double gpa = double.Parse(Console.ReadLine());

            return new Student(sid, name, department, gpa);

        }

        static List<Student> NhapDS\_List()

        {

            Console.Write("Nhập số lượng sinh viên: ");

            int n = int.Parse(Console.ReadLine());

            List<Student> students = new List<Student>();

            for (int i = 0; i < n; i++)

            {

                Console.WriteLine($"Nhập thông tin cho sinh viên thứ {i + 1}:");

                Student student = Nhap1SV();

                students.Add(student);

            }

            return students;

        }

        static void XuatDS\_List(List<Student> students)

        {

            foreach (Student student in students)

            {

                student.DisplayInfo();

            }

        }

        static void XuatDS\_ArrayList(ArrayList students)

        {

            foreach (Student student in students)

            {

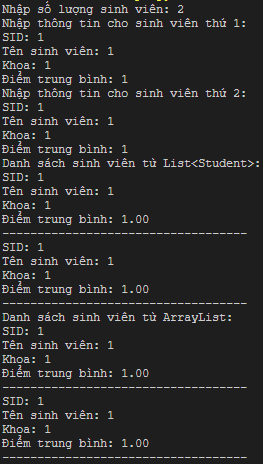
                student.DisplayInfo();

            }

        }

    }

}



Ex6

namespace Lab3

{

    public class Ex6

    {

        public static void run()

        {

            List<CanBo> danhSachCanBo = new List<CanBo>();

            bool tiepTuc = true;

            while (tiepTuc)

            {

                Console.WriteLine("Chọn loại cán bộ: 1. Nhà khoa học 2. Nhà quản lý 3. Nhân viên phòng thí nghiệm 4. Thoát");

                int luaChon = int.Parse(Console.ReadLine());

                switch (luaChon)

                {

                    case 1:

                        NhaKhoaHoc nhaKhoaHoc = new NhaKhoaHoc();

                        NhapThongTin(nhaKhoaHoc);

                        Console.Write("Nhập số bài báo đã công bố: ");

                        nhaKhoaHoc.SoBaiBao = int.Parse(Console.ReadLine());

                        danhSachCanBo.Add(nhaKhoaHoc);

                        break;

                    case 2:

                        NhaQuanLy nhaQuanLy = new NhaQuanLy();

                        NhapThongTin(nhaQuanLy);

                        danhSachCanBo.Add(nhaQuanLy);

                        break;

                    case 3:

                        NhanVienPhongThiNghiem nhanVien = new NhanVienPhongThiNghiem();

                        NhapThongTin(nhanVien);

                        Console.Write("Nhập lương tháng: ");

                        nhanVien.LuongThang = double.Parse(Console.ReadLine());

                        danhSachCanBo.Add(nhanVien);

                        break;

                    case 4:

                        tiepTuc = false;

                        break;

                    default:

                        Console.WriteLine("Lựa chọn không hợp lệ!");

                        break;

                }

            }

            XuatDanhSach(danhSachCanBo);

            TinhTongLuong(danhSachCanBo);

        }

        public static void NhapThongTin(CanBo canBo)

        {

            Console.Write("Nhập họ tên: ");

            canBo.HoTen = Console.ReadLine();

            Console.Write("Nhập năm sinh: ");

            canBo.NamSinh = int.Parse(Console.ReadLine());

            Console.Write("Nhập bằng cấp: ");

            canBo.BangCap = Console.ReadLine();

            Console.Write("Nhập chức vụ: ");

            canBo.ChucVu = Console.ReadLine();

            Console.Write("Nhập số ngày công trong tháng: ");

            canBo.SoNgayCong = int.Parse(Console.ReadLine());

            Console.Write("Nhập bậc lương: ");

            canBo.BacLuong = double.Parse(Console.ReadLine());

        }

        public static void XuatDanhSach(List<CanBo> danhSach)

        {

            Console.WriteLine("\nDanh sách cán bộ:");

            foreach (var canBo in danhSach)

            {

                Console.WriteLine(canBo);

            }

        }

        public static void TinhTongLuong(List<CanBo> danhSach)

        {

            double tongLuongNhaKhoaHoc = 0;

            double tongLuongNhaQuanLy = 0;

            double tongLuongNhanVien = 0;

            foreach (var canBo in danhSach)

            {

                if (canBo is NhaKhoaHoc)

                    tongLuongNhaKhoaHoc += canBo.TinhLuong();

                else if (canBo is NhaQuanLy)

                    tongLuongNhaQuanLy += canBo.TinhLuong();

                else if (canBo is NhanVienPhongThiNghiem)

                    tongLuongNhanVien += canBo.TinhLuong();

            }

            Console.WriteLine($"\nTổng lương nhà khoa học: {tongLuongNhaKhoaHoc}");

            Console.WriteLine($"Tổng lương nhà quản lý: {tongLuongNhaQuanLy}");

            Console.WriteLine($"Tổng lương nhân viên phòng thí nghiệm: {tongLuongNhanVien}");

        }

    }

}

A screenshot of a computer

Description automatically generated