EJERCICIOS RESUELTOS Y PROPUESTOS

Ejemplo: Gestión de Estudiantes en una Clase

Imaginemos que tenemos una aplicación para gestionar los estudiantes en una clase. Cada estudiante tiene un **nombre**, **edad** y **nota**. Usaremos la clase genérica List<T> para almacenar y gestionar los estudiantes.

Paso1: Creamos la clase Estudiante

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

namespace ConsoleApp1
{
    14 referencias
    internal class Estudiante
    {
        // Propiedades de la clase Estudiante
        4 referencias
        public string Nombre { get; set; }
        2 referencias
        public int Edad { get; set; }
        4 referencias
        public double Nota { get; set; }

        // Constructor
        4 referencias
        public Estudiante(string nombre, int edad, double nota)
        {
            Nombre = nombre;
            Edad = edad;
            Nota = nota;
        }

        // Método para mostrar la información del estudiante
        1 referencia
        public void MostrarInformacion()
        {
            Console.WriteLine($"Nombre: {Nombre}, Edad: {Edad}, Nota: {Nota}");
        }
    }
}
```

Paso2: Creamos la clase "gestionClase"

```
using System;
using System.Collections.Generic;

namespace ConsoleApp1
{
    3 referencias
    internal class gestionClase
    {
        // Lista para almacenar los estudiantes
        private List<Estudiante> estudiantes;

        // Constructor: Inicializa la lista de estudiantes
        1 referencia
        public gestionClase()
        {
            estudiantes = new List<Estudiante>();
        }
        // Método para agregar un estudiante
        4 referencias
        public void AgregarEstudiante(Estudiante estudiante)
        {
            estudiantes.Add(estudiante);
            Console.WriteLine($"Estudiante {estudiante.Nombre} agregado.");
        }
        // Método para eliminar un estudiante por nombre
        1 referencia
        public void EliminarEstudiante(string nombre)
        {
            Estudiante estudiante = estudiantes.Find(e => e.Nombre == nombre);

            if (estudiante != null)
            {
                estudiantes.Remove(estudiante);
                Console.WriteLine($"Estudiante {nombre} eliminado.");
            }
            else
            {
                Console.WriteLine($"Estudiante {nombre} no encontrado.");
            }
        }
        // Método para mostrar todos los estudiantes
        3 referencias
        public void MostrarEstudiantes()
        {
            Console.WriteLine("\nLista de Estudiantes:");
            foreach (Estudiante estudiante in estudiantes)
            {
                estudiante.MostrarInformacion();
            }
        }
        // Método para ordenar los estudiantes por nota (de mayor a menor)
        1 referencia
        public void OrdenarPorNota()
        {
            estudiantes.Sort((e1, e2) => e2.Nota.CompareTo(e1.Nota));
            Console.WriteLine("\nEstudiantes ordenados por nota de mayor a menor.");
        }

    }
}
```

Paso3: Crear el Programa Principal

```
using System;
using System.Collections.Generic;

namespace ConsoleApp1

{
    0 referencias
    class Program
    {
        0 referencias
        static void Main(string[] args)
        {
            // Crear una instancia de la gestión de clase
            gestionClase gestionClase = new gestionClase();

            // Crear algunos estudiantes
            Estudiante est1 = new Estudiante("Ana", 20, 8.5);
            Estudiante est2 = new Estudiante("Luis", 22, 9.1);
            Estudiante est3 = new Estudiante("Maria", 19, 7.3);
            Estudiante est4 = new Estudiante("Carlos", 21, 8.9);

            // Agregar estudiantes a la clase
            gestionClase.AgregarEstudiante(est1);
            gestionClase.AgregarEstudiante(est2);
            gestionClase.AgregarEstudiante(est3);
            gestionClase.AgregarEstudiante(est4);

            // Mostrar la lista de estudiantes
            gestionClase.MostrarEstudiantes();

            // Ordenar los estudiantes por nota
            gestionClase.OrdenarPorNota();

            // Mostrar la lista de estudiantes ordenados
            gestionClase.MostrarEstudiantes();

            // Eliminar un estudiante
            gestionClase.EliminarEstudiante("Maria");

            // Mostrar la lista después de eliminar a un estudiante
            gestionClase.MostrarEstudiantes();
        }
    }

}
```

EJERCICIO DE PRACTICA

Descripción del Ejercicio

Crea un programa que gestione los productos de una tienda. Cada producto tendrá las siguientes propiedades:

- **Nombre:** El nombre del producto.
- **Precio:** El precio del producto.

- **Cantidad en stock:** La cantidad de unidades disponibles en la tienda.

Tu programa debe cumplir con las siguientes funcionalidades:

1. **Agregar productos:** Permitir agregar nuevos productos a la lista.
2. **Eliminar productos:** Permitir eliminar productos por su nombre.
3. **Mostrar todos los productos:** Mostrar la lista completa de productos con sus detalles.
4. **Ordenar los productos por precio:** Ordenar la lista de productos por precio, de menor a mayor.
5. **Actualizar la cantidad en stock:** Permitir actualizar la cantidad en stock de un producto en particular.
6. **Buscar un producto por nombre:** Buscar si un producto está disponible en la tienda y mostrar su información.

Restricciones

1. No se pueden agregar productos con el mismo nombre.
2. Los precios no pueden ser negativos.
3. La cantidad en stock no puede ser negativa.

Pistas para Implementación

- **Clase Producto:** Deberá tener las propiedades Nombre, Precio, y CantidadStock.
- **Clase GestionTienda:** Deberá contener una lista genérica de productos y los métodos para realizar las operaciones solicitadas.

- **Validaciones:** Al agregar un producto, verifica que no exista otro con el mismo nombre. Asegúrate de validar que el precio y la cantidad en stock no sean negativos.