Proyección Orientada a Objetos

Teórico

¿Qué es un programa?

Conjunto de instrucciones que siguen un ordenador para realizar una actividad

¿Que es una variable?

Un carácter donde se almacena la **dirección de memoria** , donde se guarda un dato de cualquier tipo

¿Qué es la programación orientada a objetos?

Es un paradigma de la programación que se basa en la programación de **clases** a diferencia de la programación estructurada que se centra en funciones

¿Qué es una clase?

Es una plantilla que sirve para crear objetos ya predefinidos que poseen características similares , una clase sirve para representar conceptos o entidades (usuarios,noticias,coches,etc) en general sustantivos.Cada clase posee atributos(variables) y métodos(funciones) . Los métodos trabajan gracias a los atributos de la clase . Cada objeto creado a partir de una clase se le llama instancia de la clase

```
class Coche {

    // Atributos
    public int número_de_ruedas;
    public int litros_gasolina;
    public String modelo;

    // Funciones o métodos
    public void arrancar();
    public void acelerar();
    public void frenar();
}
```

¿Que es un objeto?

Es una unidad dentro de un programa tiene un estado y comportamiento es decir datos y tareas que son asignadas con esos datos , se pueden crear objetos a partir de una clase o crearlos a parte , las clases ya pre definen las instrucciones del objeto para usar las funciones de las mismas primero se debe crear el objeto de la clase

```
Coche coche1 = new Coche();
coche1.arrancar();
```

el coche 1 es una instancia de la clase surge de la clase principal

¿Qué son las listas?

las listas son un tipo de variable que nos permite almacenar un secuencia de variables tiene el mismo funcionamiento de un array se inicia desde la posición 0 .Para hacer esto debemos realizar la siguiente sintaxis

```
List<tipo de dato> nombre-lista = new List<tipo de dato>();
```

luego para agregar elementos a la lista debemos hacerlo mediante la sintaxis de

```
nombre-lista.Add(valor);
```

si queremos añadir elementos en una posición específica

```
nombre-lista.Insert(N°posición, valor);
```

si queremos eliminar elementos de una lista podemos hacerlo de 2 formas la primera mediante

```
nombre-lista.Remove(valor);
```

va a eliminar el elemento de la lista que cumpla con esta condición

y la segunda mediante

```
nombre-lista.Remove(N°posición);
```

que va a eliminar el elemento que se encuentre en la posición señalada

para ordenar elementos usamos

```
nombre-lista.Sort();
```

que lo va a ordenar de manera ascendente

¿Qué es un Constructor?

Es un código que sirve para inicializar el objeto y establecer sus propiedades y valores predeterminados. Tiene el mismo nombre que la clase y no cuenta con ningún valor de retorno, ya que su función principal es inicializar el objeto y no devolver ningún valor

```
class Person
{
    private String v1;
    private String v2;

    public Person(String v1, String v2)
    {
        this.v1 = v1;
        this.v2 = v2;
    }
}
```

¿Qué es la Sobrecarga de métodos?

Es definir dos o más métodos con el mismo nombre, pero que difieren en cantidad o tipo de parámetros ?

¿Qué es la PUBLIC?

public void se utiliza para declarar un método que es público y no estático. Esto significa que el **método es accesible desde cualquier parte del programa y puede ser llamado por cualquier objeto que tenga acceso a la clase que contiene el método**. Los métodos que no son estáticos requieren una instancia de la clase para ser creada y ejecutada.

por ejemplo persona.metodo() una instancia de la clase hace referencia al metodo

¿Qué es la STATIC?

el término "estático" se utiliza para referirse a elementos de una clase que pertenecen a la clase en sí misma, en lugar de pertenecer a una instancia de la clase. Los elementos estáticos se declaran utilizando la palabra clave static.

es decir funciones que no necesitan estar acompañadas de una instancia de la clase para ser usadas

```
public persona buscar(cosa a buscar , lista){
for lista.count
if list[i].name.toupper() == cosabuscar.toupper()
        return
```

eliminar y actualizar usamos un variable global
que ya posee los datos de la lista
previamente en el buscar

```
actualizar (persona usuario , datos)
usuario.name = dato1 ;
```

Práctico

Foreach

En C# la declaración del bucle foreach se usa de la siguiente manera

foreach (Tipodedato nombrearray in list)

Método buscar

```
static objeto buscar(int dni)
            {
                foreach (Persona p in list)
                {
                    if (p.dni == dni)
                        return p;

                }
                return null;
            }
```

Método Mostrar

```
 static Persona mostrar()
            {
                foreach (Persona p in list)
```

```
                {

                    Console.WriteLine(p.name  + " " + p.dni );
                }
                return null;
            }
```

Método Eliminar

```
  static void borrar(Persona persona1)
                {

                    list.Remove(persona1);
                }
```

Array

```
string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
cars[0] = "Opel";
Console.WriteLine(cars[0]);

// Now outputs Opel instead of Volvo
```

declarar primero el tipo le agregas corchetes de array y luego insertas los valores

Puntero

Un “puntero” es una dirección de memoria. Normalmente un puntero tendrá un tipo de datos asociado
mi puntero es persona1 por ejemplo

```
FUNCION MODIFICAR

static void modificar (Persona usuario , string nuevo nombre ){
usuario.nombre = nuevonombre;
}

funcion listar
static void listar(list)
foreach( persona p in list)


los buscadores siempre seran
```

public objeto buscar

por que estamos usando al objeto para que nos devuelva un return un valor de verdadero o falso necesitamos que devuelva un return

cuando declaró

persona persona1 → es una variable global