

INSTITUTO TECNICO INDUSTRIAL PASCUAL BRAVO

TRABAJO

La diferencia entre un objeto, clase y método

REALIZADO POR

Daniel Pino Mora

PROFESOR

Paola Andrea Quintero

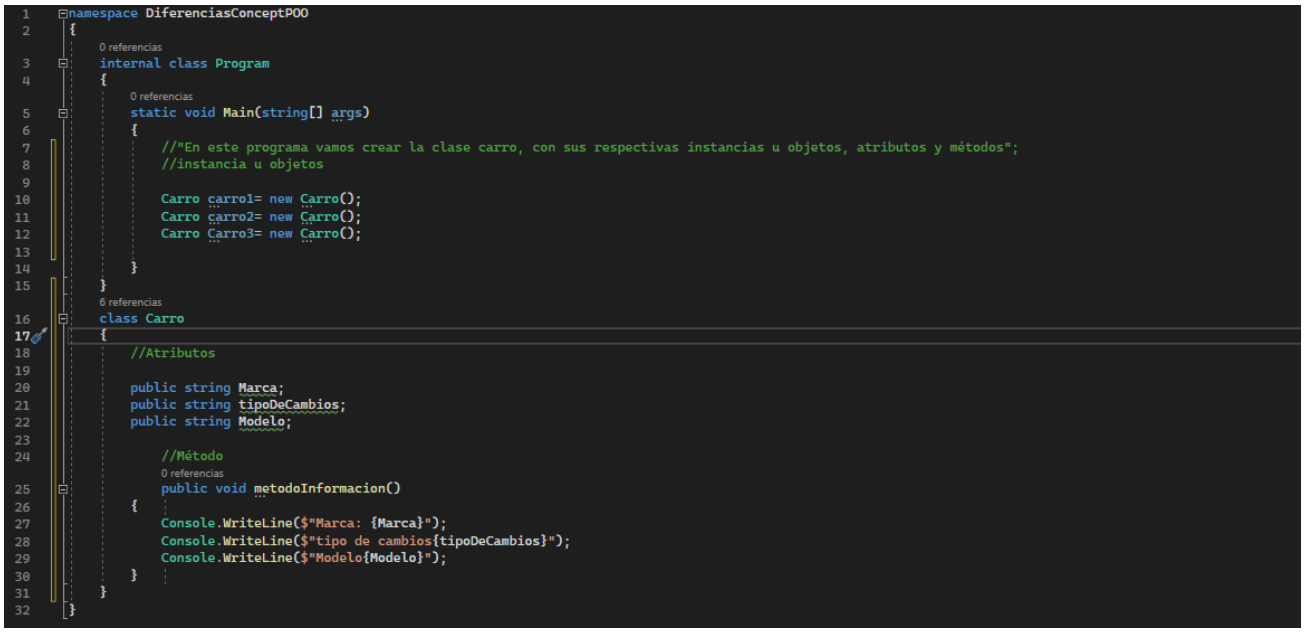
MEDELLÍN

2023

Propósito: El objetivo de este documento es demostrar mi comprensión de los conceptos de Programación Orientada a Objetos (POO), clases, instancias u objetos, atributos y métodos. Presentaré un ejemplo cotidiano que he formulado personalmente para asegurarme de que he asimilado correctamente estos temas.

Clases: Imaginemos que se establece una fábrica de automóviles; la clase abarcaría todos los automóviles en general.
Instancia u objeto: Se refiere a un automóvil individual. Por ejemplo, podríamos tener modelos como Audi, Renault, Alfa Romeo, Chevrolet y Ford.
Atributos: Los atributos representarían las características distintivas de cada uno de los objetos o marcas de automóviles. Entre estos se incluyen detalles como si es manual o automático, el número de puertas, el color (blanco, rojo, negro), si está asegurado o no, si es eléctrico, entre otros.
Métodos: Son las acciones que cada objeto puede llevar a cabo de manera individual. Por ejemplo, los automóviles de cierta marca podrán frenar, desplazarse, emitir señales de radio y Bluetooth. En contraste, otros no podrán emitir señal Bluetooth, pero podrán rastrear con GPS y realizar frenadas de emergencia, entre otras funcionalidades.

Código y explicación



```
namespace DiferenciasConceptPOO
{
    0 referencias
    internal class Program
    {
        0 referencias
        static void Main(string[] args)
        {
            //"En este programa vamos crear la clase carro, con sus respectivas instancias u objetos, atributos y métodos";
            //instancia u objetos

            Carro carro1= new Carro();
            Carro carro2= new Carro();
            Carro Carro3= new Carro();

        }
    }
    6 referencias
    class Carro
    {
        //Atributos

        public string Marca;
        public string tipoDeCambios;
        public string Modelo;

            //Método
            0 referencias
            public void metodoInformacion()
        {
            Console.WriteLine($"Marca: {Marca}");
            Console.WriteLine($"tipo de cambios{tipoDeCambios}");
            Console.WriteLine($"Modelo{Modelo}");
        }
    }
}
```

Para crear una clase, debemos definirla fuera del método Main y fuera de la clase principal del programa, pero dentro del mismo namespace. Los objetos de esa clase se crean dentro del método Main.

`class Carro` Creamos la clase carro luego de crearla debomos de agregar doble llave {}

```
//Atributos

public string Marca;
public string tipoDeCambios;
public string Modelo;
```

Los atrributos se crean por dentro de las llaves {} de la clase en este caso sus atributos son publicos

```
    //Método
    0 referencias
    public void metodoInformacion()
{
    Console.WriteLine($"Marca: {Marca}");
    Console.WriteLine($"tipo de cambios{tipoDeCambios}");
    Console.WriteLine($"Modelo{Modelo}");
}
```

Las acciones de la clase o sea el método o los métodos, también se crean dentro de la llaves {} de la clase

```
0 referencias
static void Main(string[] args)
{
    //"En este programa vamos crear la clase carro, con sus respectivas instancias u objetos, atributos y métodos";
    //instancia u objetos

    Carro carro1= new Carro();
    Carro carro2= new Carro();
    Carro Carro3= new Carro();

}
```

Por último creamos los objetos dentro de la clase del método Main con la nomenclatura que se ve en la imagen

Conclusión:

En este documento, he abordado los fundamentos de la Programación Orientada a Objetos (POO), explorando conceptos esenciales como clases, instancias u objetos, atributos y métodos. Mi objetivo principal ha sido presentar un ejemplo concreto y de la vida cotidiana, formulado por mí, para asegurarme de que he adquirido un entendimiento sólido de estos conceptos fundamentales.
En el contexto de la programación orientada a objetos, las clases se conciben como plantillas o modelos que definen la estructura y el comportamiento compartido por un grupo de objetos relacionados. Utilizando el ejemplo de una fábrica de automóviles, hemos ilustrado cómo una clase puede abarcar todos los automóviles en general, proporcionando una base común para crear instancias u objetos individuales, como Audi, Renault, Alfa Romeo, Chevrolet y Ford.
Cada una de estas instancias representa un automóvil específico y único, y se identifica por sus atributos distintivos. Los atributos, que abarcan desde detalles mecánicos como el tipo de transmisión y el número de puertas hasta características visuales como el color, permiten que cada automóvil tenga su propia identidad. Además, los atributos pueden influir en los métodos que un automóvil puede ejecutar de manera individual.
Los métodos, por su parte, representan las acciones que un automóvil puede llevar a cabo de forma autónoma. En nuestro ejemplo, hemos demostrado cómo ciertos automóviles pueden frenar, desplazarse y emitir señales de radio y Bluetooth, mientras que otros poseen capacidades adicionales como rastreo con GPS y frenadas de emergencia.