



OOP C#

Definizione OOP

- Un programma è costituito da oggetti che interagiscono tra loro.
- Il dato viene visto come un tipo di dato astratto (ADT) caratterizzato da un insieme di valori che lo definiscono e un insieme di operazioni che possono essere applicate a esso.
- Modularità e riusabilità del software
- Facilita la gestione e la manutenzione di progetti.
- Qualsiasi modello include gli aspetti più importanti o essenziali di qualcosa mentre ignora i dettagli meno importanti.
- Enfatizzare gli aspetti comuni.

Oggetti

- Gli oggetti sono entità (reali o astratte) che troviamo nel mondo reale che ci circonda.
- Componenti hardware/software, documenti, esseri umani, veicoli, alberi, ogni elemento del mondo può essere rappresentato attraverso un oggetto.

Navigatore

Mike è un programmatore che deve andare al colloquio per un nuovo posto.

Non vede l'ora di mostrare le sue capacità di C#, ma prima deve arrivarci, ed è in ritardo.

Mike si immagina quali vie percorrere per raggiungere il luogo (prendo A2 direzione Sud, esco a Mendrisio e poi vado su via Vignalunga – definisce la destinazione ottenendo il percorso)

Per fortuna ha una radio che annuncia dei forti rallentamenti sulla A2 sul ponte di Melide. Mike ha una nuova informazione sulla strada da evitare.

Mike sceglie una via alternativa (strada cantonale) per raggiungere il luogo dell'appuntamento in orario.



Modelliamo la classe navigatore

Classe Navigatore

La classe navigatore ha metodi per definire e modificare le strade.

La classe ha metodi nelle quali avvengono le azioni.

Ma a differenza del metodo `button_Click()`, essi sono predisposti per un unico problema: trovare percorsi in un paese.

Per questo Mike li ha raccolti tutti in una classe chiamata Navigatore.

Mike ha disegnato la sua classe Navigatore in modo da definire o modificare facilmente le strade.

Per trovare un percorso, il programma chiama il metodo `InserireDestinazione()` e usa `TrovarePercorso()` per ottenerlo in una stringa.

Se deve modificare il percorso, richiama `InserireDeviazione()` e poi richiama di nuovo `TrovarePercorso()` per ottenere le nuove indicazioni.

Navigatore
+Marca : string +Dimensioni : int +Modello : string
+InserireDestinazione() +TrovarePercorso() : string +InserireDeviazione() +DistanzaTotale() : int +TempoTotale() : double

Implementare la classe navigatore

Struttura di una classe

Di seguito è presente la sintassi per creare una classe in C#.

```
public class Navigatore{  
    // dichiarazione di attributi, metodi e costruttori  
}
```

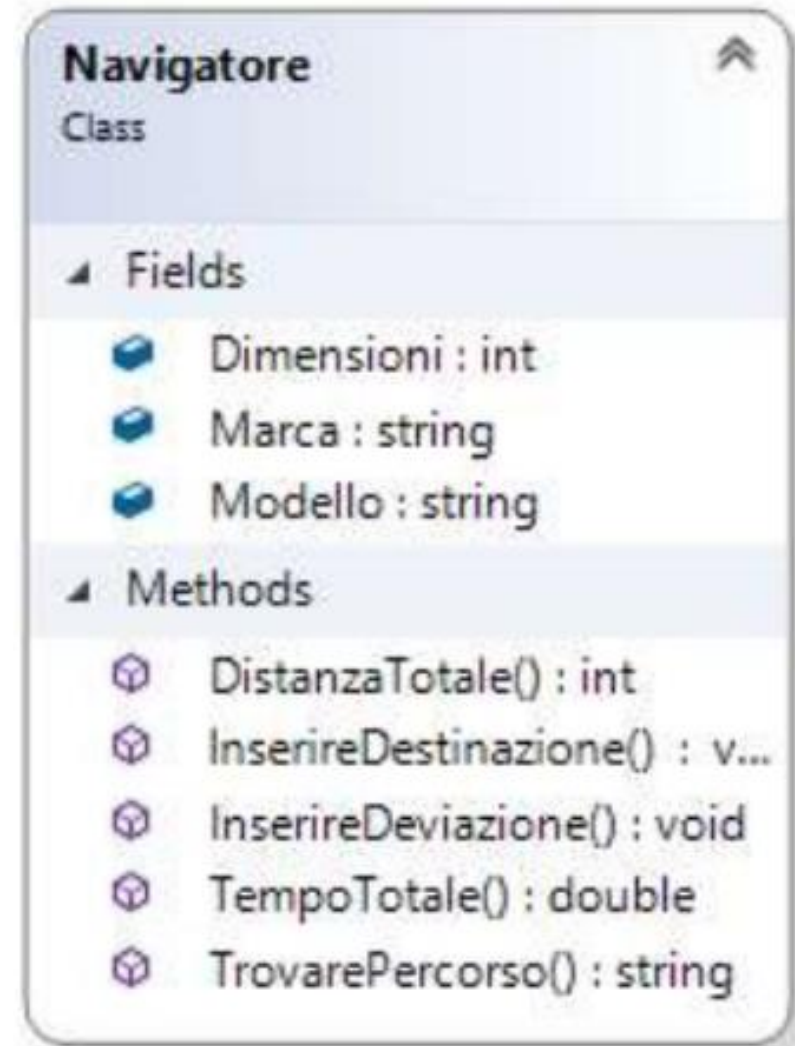
Come accennato durante i capitoli introduttivi, il nome della classe per convenzione inizia con una lettera in maiuscolo.

Creare una classe

All'interno del blocco (parentesi graffe) della classe è presente tutto il codice che serve per il ciclo di vita degli oggetti (**istanze**).

```
public class Navigatore
{
    public string Marca;
    public int Dimensioni;
    public string Modello;

    public string TrovarePercorso(){
        //implementare metodo
    }
    public void InserireDeviazione(string evita){
        //implementare metodo
    }
    public int DistanzaTotalei(){
        //implementare metodo
    }
    public double TempoTotale(){
        //implementare metodo
    }
}
```



Metodi di istanza

I metodi di un oggetto (detti anche metodi di istanza) **non sono dichiarati** **static** e hanno la facoltà di accedere ai valori degli attributi.

In questo modo possono modificare lo stato dell'oggetto, oppure modificare il proprio comportamento in base ad esso.

```
public int DistanzaTotale(){  
    return 55;  
}
```

Istanze

Una volta definite le classi utili per realizzare la logica di un programma, si possono istanziare degli oggetti secondo il paradigma OOP.

Program.cs

```
internal class Program
{
    static void Main(string[] args)
    {
        string arrivo;
        Navigatore tomTom;
        tomTom = new Navigatore();
        arrivo = "Lugano";
        Console.WriteLine(tomTom.trovarePercorso(arrivo));
        Console.ReadKey();
    }
}
```

Creazione di oggetti – Istanziamento

Nella programmazione orientata agli oggetti una classe non può essere utilizzata direttamente, ma è necessario un oggetto o istanza di quella classe. In Java la creazione di un oggetto avviene in due fasi:

- Dichiarazione di un oggetto
- Allocazione di un oggetto

```
Navigatore tomTom = new Navigatore();
```



Nome classe

Nome variabile

Nome Metodo Costruttore