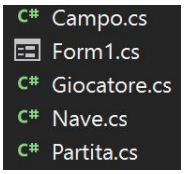
Progetto: Battaglia Navale

Classi:



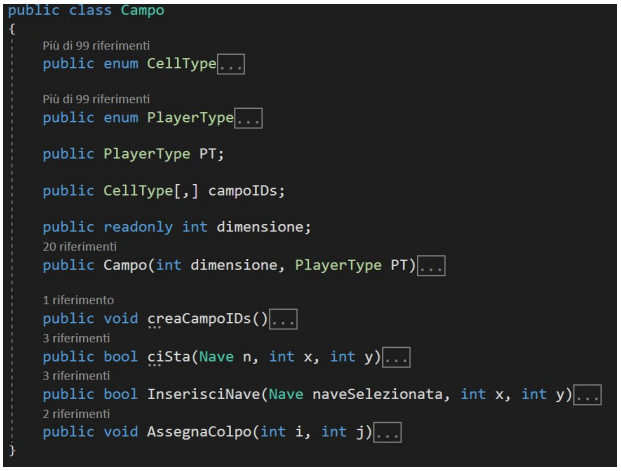
Classe **Partita**:

```
internal class Partita
{
    private const int dimensione = 6;
    public int nNavi = 0;

    public Campo campo = new Campo(dimensione, Campo.PlayerType.Io);
    public Campo campoAvversario = new Campo(dimensione, Campo.PlayerType.Nemico);

    private Giocatore g1;
    private Giocatore g2;

    public Giocatore[] giocatori = new Giocatore[2];

    1 riferimento
    public Partita(Giocatore g1, Giocatore g2)
    {
        this.g1 = g1;
        this.g2 = g2;

        giocatori[0] = g1;
        giocatori[1] = g2;

        g1.creaNaviPosizionabili(dimensione);
        g2.creaNaviPosizionabili(dimensione);
        nNavi = g1.navi.Count;
    }
}
```

- **Costruttore**: Partita(Giocatore g1, Giocatore g2);

- **Attributi**:
  - Campo campo: campo del giocatore umano,
  - Campo campoAvversario: campo del computer;
  - Giocatore[] giocatori: vettore contenente i due giocatori.

Classe **Campo**:



- **Costruttore**: Campo(int dimensione, PlayerType PT), crea il campo in base alla dimensione desiderata e in base al tipo di giocatore a cui assegnare il campo.
    - Enum PlayerType:

```
public enum PlayerType
{
    Io, Nemico
}
```

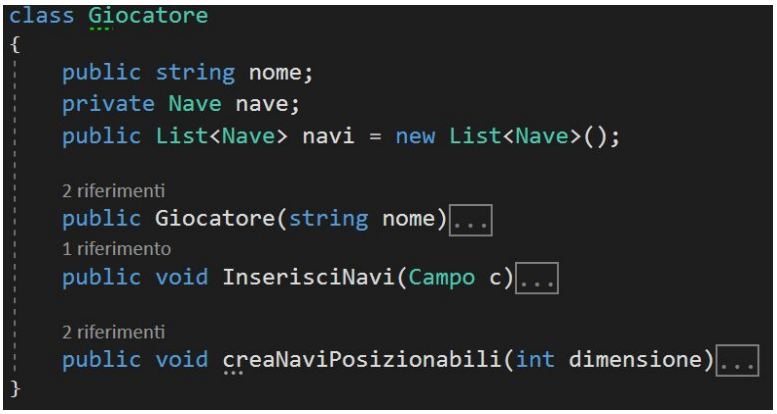
- **Attributi**:
    - Enum CellType: tipo di cella nel campo

```
public enum CellType
{
    Mare, Buco, Colpito, Affondato, Barca
}
```

    - Enum PlayerType: tipi di player
    - PlayerType PT: tipo di campo in base al tipo di player
    - CellType[,] campoIDs: campo di id.
    - Int dimensione: dimensione del campo

- **Metodi**:
    - creaCampiIDs(): crea il campo inizializzato con inizialmente solo mare
    - Bool ciSta(Nave n, int x, int y): ritorna se una nave ci sta in una posizione x, y;
    - Bool InserisciNave(Nave n, int x, int y): inserisce la nave e ritorna se l'inserimento è riuscito;
    - AssegnaColpo(int i, int j): cambia l'id della cella in quelle coordinate

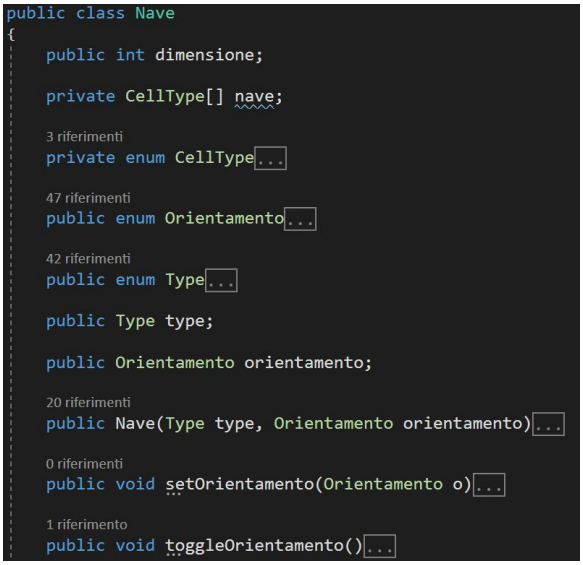
Classe **Giocatore**:



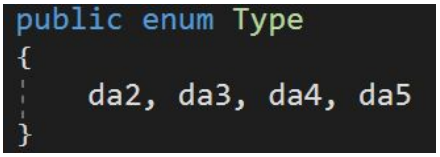
- **Costruttore**: Giocatore(string nome): inserisci nome
- **Attributi**:
    - String nome: nome del giocatore;
    - List<Nave> navi: navi del giocatore

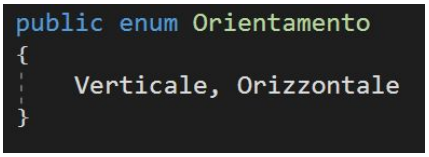
- **Metodi**:
    - InserisciNavi(Campo c): inserimento navi casuale;
    - creaNaviPosizionabili(int dimensione): crea le navi che possono essere inserite da un giocatore

Classe **Nave**:

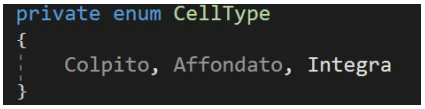


- **Costruttore**: Nave(Type type, Orientamento orientamento): la nave ha un tipo (la lunghezza) e un orientamento (verticale od orizzontale)





- **Attributi**:
  - Int dimensione: dimensione della nave
  - CellType[] nave: la nave è un vettore di tipo di cella, che può essere integra, colpita o affondata



  - Enum Orientamento: la nave può essere orizzontale o verticale

- **Metodi**:
  - setOrientamento(Orientamento o): setta l'orientamento
  - toggleOrientamento(): toggla l'orientamento