

Test di fine settimana – Week 4

Nome	Michela
Cognome	Murtas
Data	23/07/2021

Leggete attentamente ogni domanda e argomentare quanto più possibile **fornendo anche degli esempi**.
ATTENZIONE: Le domande a risposta multipla possono contenere più risposte corrette.

1. *Dare una definizione di classe statica*
 - ✓ La classe statica è una classe nella quale non è possibile istanziare (ovvero creare) oggetti.
2. *Nell'istruzione **Persona p = new Dipendente();** la variabile p si comporta in maniera polimorfa. Perché? Cosa si intende per comportamento polimorfico?*
 - ✓ Il Polimorfismo è capacità di trattare un'istanza di un dato tipo come se fosse di un altro tipo. In questo caso Persona P è in tutto e per tutto un Dipendente perché prenderà tutto quello che è contenuto in Dipendente.
3. *In che modo è possibile utilizzare la combinazione di keyword del linguaggio virtual e override? Fornire un esempio.*
 - ✓ Con Virtual si concede la possibilità alle classi derivate di implementare un metodo ereditato attraverso l'utilizzo dell'Override.
Esempio, riprendendo Persona e Dipendente.
Su Persona potremo avere un metodo ToString:

```
public virtual string Abitazione(string città)
{
    return $"Abita a {città}";
}
```


Su Dipendente ci sarà l'override nel quale posso agire in tre modi diversi.
 1. Lo lascio così com'è.

```
public override string Abitazione(string città)
{
    return base.Abitazione(città);
}
```
 2. Lo ridefinisco

```
public override string Abitazione(string città)
{
    return $"{{Nome}}{{Cognome}} vive a {città}";
}
```
 3. Aggiungo qualcosa

```
public override string Abitazione(string città)
{
    return base.Abitazione($"{{città}} in via {{Indirizzo}}");
}
```

4. *Spiegare il significato di Tipo, Classe e Istanza nella programmazione OOP*
 - ✓ *Per Tipo si intende rappresentazione di un oggetto in modo concreto attraverso le Classi.*
 - ✓ *Le Classi infatti sono dei tipi definiti dall'utente che vengono utilizzate per creare degli oggetti.*
 - ✓ *Gli Oggetti sono le istanze di una Classe, che sono caratterizzate da un'identità, un comportamento (i metodi) e uno stato.*

5. *Quali tipologie di ereditarietà sono consentite in C# e come si definisce? Fornire un esempio*
 - ✓ *Su C# è consentita SOLO l'ereditarietà singola, quindi una classe può derivare solo ed esclusivamente da una classe (in C# una Matita può derivare da Cancelleria, ma non può derivare ANCHE da OggettiInLegno)*

Esercitazione pratica

Realizzare una Console Application per gestire una squadra di calcio.

La squadra è caratterizzata dai seguenti attributi:

- *Nome*
- *Data di Fondazione*
- *Campionato disputato*
- *Nome del Presidente*
- *Nome dell'Allenatore*
- *Rosa (elenco di Calciatori)*

Ciascun Calciatore in Rosa appartiene ad una categoria: Portiere, Difensore, Centrocampista, Attaccante.

Tutti i giocatori sono in possesso dei seguenti attributi:

- *Numero di Maglia*
- *Nome*
- *Data di Nascita*

Inoltre:

- *Per i Portieri vanno registrati il Numero di Gol Subiti e il Numero di Rigori Parati*
- *Per i Difensori vanno registrati il Numero di Tackle Riusciti*
- *Per i Centrocampisti vanno registrati il Numero di Passaggi Tentati e il Numero di Passaggi Riusciti*
- *Per gli Attaccanti vanno registrati il Numero di Gol Realizzati*

L'applicazione deve permettere di:

- *Aggiungere Giocatori alla Rosa*
- *Vendere Giocatori (rimuoverli dalla Rosa)*
- *Gestire la squadra titolare (ovvero identificare gli 11 giocatori titolari) con lo schema 4-4-2 (4 Difensori, 4 Centrocampisti, 2 Attaccanti)*
- *Stampare le statistiche di tutta la squadra titolare*

Mettere la prova pratica e teorica su Github.