

Test di fine settimana – Week 4

Nome	Laura
Cognome	Martines
Data	23 luglio 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 keyword static si usa per marcare membri di una classe che esistono indipendentemente dall'istanza (quindi, ai quali si può accedere senza passare attraverso una istanza della classe, in quanto vi si accede direttamente usando il nome della classe). Essi descrivono informazioni comuni a tutte le istanze e vengono detti membri di classe (a differenza dei membri non statici, che vengono detti membri di istanza). È consigliabile rendere statiche le classi che contengono solamente membri statici. Quando una classe viene definita statica, non è possibile creare un'istanza della classe, quindi non si può usare la keyword new.

2. *Nell'istruzione **Persona p = new Dipendente();** la variabile p si comporta in maniera polimorfa. Perché? Cosa si intende per comportamento polimorfo?*

Nell'istruzione di esempio, la variabile p si comporta in modo polimorfo perché essa è una istanza della classe Dipendente (quindi ha associati tutti i membri di tale classe) ma viene "trattata" come una istanza della classe Persona. Per poter fare ciò, è necessario che Persona sia la classe base di Dipendente, ossia che Dipendente erediti da Persona. La conseguenza di questa assegnazione è che diventa possibile accedere solamente ai membri di Dipendente che sono presenti anche in Persona, mentre quelli propri della classe Dipendente (sebbene presenti) non sono direttamente accessibili, ma per visualizzarli è necessario fare un cast. Se però nella classe base sono presenti metodi di cui è stato fatto un override nella classe Dipendente, se vengono richiamati per l'istanza p, essi eseguono le operazioni specificati nella classe Dipendente.

3. *In che modo è possibile utilizzare la combinazione di keyword del linguaggio virtual e override? Fornire un esempio.*

Le keyword virtual e override si usano in combinazione per eseguire l'override di un metodo della classe madre nelle classi derivate. In particolare, nella classe base si indica che il metodo può essere modificato nelle classi dipendenti tramite la keyword virtual, mentre per indicare nelle classi derivate che il metodo specificato va effettivamente a modificare quello presente nella classe base, si inserisce la keyword override. Se in una classe derivata non viene fatto l'override del metodo, quando esso verrà richiamato andrà a richiamare la classe base e quindi userà quello in esso specificato.

Esempio:

```
public class Veicolo
{
    public virtual MoveTo() {}
}

Public class Aeroplano : Veicolo
{
    public override MoveTo() {}
}
```

```
Public class Macchina : Veicolo
{
}
```

In questo caso, veicolo è la classe base, mentre Aeroplano è la classe da essa derivata. Il metodo MoveTo è definito virtual in Veicolo, e questo indica che è possibile modificarlo nelle classi derivate, anche se tale modifica è appunto una possibilità, quindi opzionale. Nella classe derivata Aeroplano, è stato definito nuovamente il metodo MoveTo, ed è stato contrassegnato con la keyword override che indica che esso sostituisce quello di Veicolo. La classe Macchina invece non contiene un override del metodo MoveTo, quindi se tale metodo dovesse essere richiamato per istanze della classe Macchina, verrà eseguito quello definito nella classe Veicolo.

4. *Spiegare il significato di Tipo, Classe e Istanza nella programmazione OOP*

I tipi sono rappresentazioni concrete degli oggetti; possono essere value type o reference type e ci indicano come un oggetto viene rappresentato (se un numero viene rappresentato come intero o decimale, per esempio). Le classi sono reference type definite dall'utente, che attraverso esse può rappresentare degli oggetti associandoci campi, proprietà e metodi. Una istanza di una classe è invece un singolo oggetto appartenente alla classe, che è quindi caratterizzato dalle sue proprietà specifiche che lo distinguono dalle altre istanze della stessa classe. Per dichiarare una istanza di una classe si usa la keyword new.

5. *Quali tipologie di ereditarietà sono consentite in C# e come si definisce? Fornire un esempio*

L'ereditarietà imposta un vincolo di tipo "è un tipo di" tra classe base e derivata (la classe derivata è un tipo di classe base) e permette di estendere, specializzare o modificare il comportamento di una classe. Una classe derivata eredita tutti i membri della classe base, ma con il loro access modifier (quindi non può accedere ai membri privati, ad esempio). C# consente solamente un tipo di ereditarietà, ossia quella singola: ogni classe può avere una sola classe base (quindi può ereditare da una sola classe), sebbene ogni classe possa essere la classe base di infinite classi. Quindi una classe può avere una sola classe base, ma ogni classe base può avere infinite classi derivate. Per bypassare i limiti dati da ciò, è possibile utilizzare le interfacce. Per indicare che una classe eredita da un'altra, quando si definisce una classe, si mette accanto al suo nome la dicitura : "Nome della classe base"

Ad esempio,

```
Class Veicolo {}
Class Aeroplano : Veicolo {}
```

Indica che la classe base è Veicolo, mentre Aeroplano eredita da Veicolo, è una sua classe derivata.

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.