# Test di fine settimana – Week 4

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Roberta |
|  |  | Cognome | Beretta |
|  |  | 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*

*Una classe è un oggetto che rappresenta un tipo di dato custom, cioè che è possibile definire e ritagliare su misura in base alle esigenze dello sviluppatore.*

*Statica è un aggettivo che in C# viene rappresentato dalla keyword static e risulta essere un modificatore di accessibilità.*

*Quando una classe viene marcata come static significa che i membri di tale classe sono comuni a tutta la classe, dunque non è necessario istanziare oggetti di quella classe. I membri di una classe statica sono necessariamente statici.*

*Supponiamo di avere una classe che si chiama Triangolo che risulta essere statica.*

*Supponiamo che in tale classe si abbia un metodo chiamato Area che calcola appunto l’area di un triangolo prendendo in input base e altezza del triangolo.*

*Allora per utilizzare tale metodo si usa la classe stessa facendo Triangolo.Area( baseTriangolo , altezzaTriangolo).*

*Public static float Area(float base, float altezza){*

*return baseTriangolo\*altezzaTriangolo/2;}*

*Se la classe non fosse stata statica e nemmeno il metodo Area fosse stato statico, avrei potuto istanziare un oggetto triangolo di tipo Triangolo con la proprietà di avere una certa Base ed una certa Atezza assegnate, ad esempio, al momento della dichiarazione. Poi avrei calcolato l’area facendo triangolo.Area().*

*Public float Area()*

*return triangolo.Base\*triangolo.Altezza/2;}*

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

*Il Polimorfismo è uno dei tre pilastri della programmazione ad oggetti con Incapsulamento ed Ereditarietà.*

*Il Polimorfismo è la possibilità di un oggetto di assumere valori di un altro tipo di dato rispetto a quello con cui è stato dichiarato. Ovviamente non può assumere il comportamento e il valore di qualsiasi tipo di dato, ma lo può fare entro il range definito dalla gerarchia delle classi.*

*Nell’esempio abbiamo che ogni dipendente è una persona ma non tutte le persone sono dipendenti.*

*Dipendente è una classe derivata dalla classe base Persona.*

*Dichiarando dipendente come fatto nella domanda io sto allocando nello heap uno spazio di tipo Dipendente che però, almeno momentaneamente, decido di guardare nello stack come una Persona. Dunque non vedrò le caratteristiche proprie di un dipendente come ad esempio la proprietà TipoDiContratto che è specifica del dipendente ma non di tutte le Persone, perché momentaneamente sono nascoste.*

*Se voglio però, attraverso una istruzione di casting posso avere quelle proprietà e quei comportamenti tipici del Dipendente perché al momento della dichiarazione, nonostante abbia tipizzato il dato come Persona, ho creato effettivamente in memoria una nuova istanza di Dipendente. Ho fatto dunque una azione di overtyping.*

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

*Per spiegare le keyword virtual ed override bisogna dire che cosa è l’Ereditarietà.*

*L’Eereditarietà è un modo di creare una gerarchia tra le classi con il constraint (in C#) che ogni classe può avere una e una classe base da cui discente. Una classe può avere più classi discendenti.*

*Supponiamo di avere la classe base Persona e la classe discendente Dipendente.*

*Supponiamo che la classe Persona abbia due metodi, uno marcato come virtual e uno senza la keyword virtual.*

*Allora la classe Dipendente erediterà entrambi i metodi quindi potrà utilizzarli entrambi, ma con una differenza.*

*Nel caso del metodo base marcato con virtual, in Dipendente io posso marcare il metodo con override e modificarlo per farlo su misura alla classe Dipendente.*

*Nell’altro caso, cioè se il metodo nella classe base non ha la keyword vistual, non lo posso fare.*

*Abbiamo visto in classe l’esempio di .ToString() che è un metodo della classe base Object marcato con virtual che possiamo personalizzare, mettendo override, nella nostra classe che discende di default da object.*

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

*Il Tipo è un tipo di dato che viene dato di default come ad esempio int per gli interi, string per le stringhe di caratteri, float per i reali, bool per i booleani cioè vero o falso.*

*Questi tipi di dato possono essere tipi di valore o di riferimento. I primi istanziano variabili o strutture dati che contengono il dato stesso mentre i secondi contengono un indirizzo che punta a dove il dato è memorizzato.*

*Le Classi sono un tipo di dato definito dallo sviluppatore ed ogni classe è di tipo riferimento.*

*Infine le istanze sono gli oggetti “materiali” che hanno un tipo o una classe.*

*Ad esempio se io dichiaro*

*int numero = 3;*

*io sto dichiarando una istanza (in questo caso una variabile) di tipo intero che si chiama numero e a cui assegno il valore 3.*

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

*Ci sono due tipologie di Ereditarietà: singola e multipla.*

*In C# non è ammessa l’ereditarietà singola perché ogni classe deve necessariamente avere una e una sola classe base da cui discende. Nel caso in cui questa richiesta non sia presente, si parla di ereditarietà multipla.*

*Nonostante abbia i suoi vantaggi permettere ad una classe di discendere da due classi distinte, sono probabilmente più gli svantaggi perché poi risulta difficile capire come gestire i metodi ereditati, infatti alcuni potrebbero anche essere in contrasto.*

*In C# tuttavia è data la possibilità attraverso le interfacce di godere di alcuni vantaggi dati dall’ereditarietà multipla.*

*Ad esempio supponiamo di avere la classi Pianoforte, Orchestra e StrumentoMusicale.*

*Allora pianoforte potrebbe ereditare sia da Orchestra che da StrumentoMusicale tuttavia ciò non è consentito ma è possibile solo che pianoforte erediti o da StrumentoMusicale oppure da Orchestra.*

*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.*