# Test di valutazione – Modulo 2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | LORELLA |
|  |  | Cognome | TUNDISI |
|  |  | Data | 18/10/2019 |

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. *Cosa si intende quando si utilizza il tipo “var” durante la dichiarazione delle variabili?*

la variabile non ha un tipo specifico quindi può assumere qualunque tipo

il motore di .NET determina il tipo a runtime time

il compilatore C# determina il tipo a compile time

la variabile ha un tipo implicito che potrebbe essere anche esplicitamente dichiarato a codice

1. *A cosa servono il Costruttore e il Distruttore di una classe?*

A creare e distruggere la classe a cui sono applicate

A gestire le operazioni di inizializzazione e pulizia delle risorse di una classe

A permettere ad una classe di gestire in maniera efficiente il Garbage Collector

A inizializzare una classe senza che la stesse venga terminata dal Garbage Collector

1. *Quali sono i pilastri sulla quale si basa la OOP? Dare una definizione di ciascuno di essi descrivendo per quale motivo sono fondamentali per questo tipo di approccio allo sviluppo*

*La programmazione orientata agli oggetti è basata su alcuni pilastri fondamentali (concetti): anzitutto è fondamentale il concetto di classe, che definisce dei tipi di dato e consente la creazione di Oggetti.*

*OGGETTI: sono istanze di una classe; ossia partendo da una classe, si può definire un oggetto che ha uno stato iniziale che, attraverso un metodo, si può modificare.*

*-EREDITARIETà : le classi che ereditano da altre classi (superclassi) ne prendono proprietà e metodi; non è concessa l’ereditarietà multipla (una classe non può ereditare da più classi, ma da 1 sola). Consente di definire quindi una classe partendo da una già creata.*

*-INCAPSULAMENTO: è la possibilità di limitare l’accessibilità delle classi agli utenti o agli sviluppi del codice per cui non sono necessari.*

*-POLIMORFISMO: può essere di 2 tipi (overriding e overloading dei metodi)e consiste nella possibilità di utilizzare lo stesso codice con istanze di classi diverse.*

*Possiamo dire che grazie a questi “pilastri” gli oggetti scambiano tra loro messaggi che consentono la creazione di un’applicazione “reale”.*

1. *Qual è la differenza tra overloading e overriding? Fornire un esempio di una classe per cui si applica un override di un metodo e un overload dello stesso metodo*

*L’overriding e l’overloading rientrano nel Polimorfismo. Il primo consiste nel poter usare un metodo che si comporta in maniera diversa in base alla classe che lo richiama (es. Le classi AUTO e BICI, derivate dalla classe MEZZI DI TRASPORTO, hanno entrambe un metodo Accelera che si comporterà in maniera diversa nelle 2 classi). Il secondo permette a una classe di avere metodi che hanno lo stesso nome ma parametri diversi (in effetti hanno una firma diversa, cioè stesso nome ma per es. Numero di parametri diversi).*

1. *Cosa è il Garbage Collector e a cosa serve nel mondo .NET?*

*Il Garbage Collector interviene quando si verifica un memory leak, ossia quando la memoria arriva a limite nell’implementazione di un’applicazione. Es: Avviando una funzione che stampa “all’infinito”su console (attraverso una funzione che non esce mai dal ciclo), la memoria sarà utilizzata fino al limite; a quel punto il Garbage Collector interviene per evitare il collasso del sistema operativo, ossia liberando memoria cercando tra gli oggetti “zombie”, ossia quelli non necessari al funzionamento dell’applicazione o vitali per il mantenimento del sistema operativo (al limite l’esecuzione dell’applicazione viene bloccata).*

*L’utilizzo della memoria può essere facilmente visibile dal Task Manager.*

1. *Qual è la differenza tra Values Type e Reference Type?*

*Una variabile Value Type contiene un valore del tipo (es. Un intero), invece una Reference Type contiene un’istanza del tipo, ossia un oggetto.*

1. *Cos’è e a cosa serve LINQ? Dare una definzione dello stesso, e descrivere per quale motivo è diventato fondamentale per aumentare la produttività di uno sviluppatore .NET*

*LINQ, o Language integrated query, è il nome di un set di tecnologie basate sull’integrazione delle funzionalità di query (SQL) direttamente nel linguaggio C#.*

*LINQ permette di fare programmazione funzionale, ossia basata sull’uso di funzioni vuote, che permettono al programmatore di arrivare ad uno stesso risultato utilizzando meno codice.*

1. *Quali sono le differenze tra una classe, una classe astratta e un’interfaccia? Elencare quante più caratteristiche possibili di ciascuno di questi costrutti*

*Una classe ha proprietà (o attributi) e metodi che la definiscono in maniera univoca.*

*La classe astratta ha dei metodi generici che possono essere richiamati dalle sottoclassi (es. Una classe astratta ha il meodo CREA a cui è passato un tipo generico). Nella definizione, si premette il termine “abstract” al nome della classe astratta.*

*Es: public abstract class ManagerBase<TEntity>*

*where TEntity: EntitaMonitorabile*

*{public void Crea(TEntity entityDaCreare)*

*…}*

*L’interfaccia si comporta come una classe astratta ma quando si difinisce non ha modificatori di accesso (public, private..). Essa è un contratto che le altre classi che la implementano devono rispettare ; essendo una generalizzazione, non è possibile fare il “new”di un’interfaccia. E’ molto utile quando si vuole fare un’ override dei metodi, ossia utilizzare un metodo generico che si comporta in maniera diversa in base alle classi da cui è richiamato: infatti utilizzare l’interfaccia -piuttosto che una classe astratta- fa in modo che si eviti il limite della ereditarietà multipla, che con le classi “normali” non è concessa (es. Una classe può ereditare da 2 interfacce, ma non può ereditare da 2 classi astratte “generiche”).*

1. Come si definisce una “classe parametrica”? Fornire un esempio della stessa, fornendo sia il codice implementativo che il suo utilizzo pratico

Immaginiamo di voler creare una libreria virtuale. Sicuramente ci servirà una funzione che andrà ad aggiungere un nuovo libro o un nuovo genere al database (per. es un semplice file .txt). Piuttosto che fare 2 funzioni “Crea” potremmo creare una classe parametrica, di nome “ManagerBase” così definita, insieme a un suo metodo “Crea”:

public abstract class ManagerBase<TEntity>

where TEntity: EntitaMonitorabile

{public void Crea(TEntity entityDaCreare)

…}

La classe parametrica è una classe astratta e serve quindi a svincolare le funzioni dal contesto in cui si applicano. Ha passato al suo interno un parametro (nell. esempio Tentity deriva da EntitaMonitorabile, che è il nome di un’entità; EntitaMonitorabile è una classe astratta da cui derivono delle classi – es.LIBRO,GENERE,UTENTE- ; ogni volta che si vuole creare un nuovo oggetto “libro”oppure “genere” si richiama il metodo “Crea”, che si comporterà in maniera diversa in base all’entità da creare -appunto un libro, un genere o un utente.

1. *Esercitazione pratica. Si chiede di creare un app console in .NET Framework che, in fase di esecuzione, permetta di inserire e visualizzare un catalogo di automobili e biciclette (entrambe le entità avranno le medesime funzioni).* 
   * *Quando l’applicazione viene avviata permette di selezione se si vuole lavorare sul catalogo delle biciclette (selezione da menu “A”) o quello delle automobili (selezione da menu “B”).*
   * *Dovrà essere possibile inserire una bicicletta (o un’automobile) e salvarla su un “database” realizzato con un file di json*
   * *Dovrà essere possibile visualizzare il contenuto intero del database dell’oggetto di riferimento*
   * *Dovrà essere possibile visualizzare solo gli elementi che corrispondono al criterio di ricerca sul campo “Modello”.*
   * *La bicicletta è definita dal campo “Modello”, “Marca”, “NumeroTelaio” e “IsElettrica”;*
   * *L’automobile è definita da “Modello”, “Marca”, “NumeroCavalli”, “IsDiesel” e “DataImmatricolazione”.*
   * *Ogni entità è caratterizzata da un campo “Id” numerico che deve essere univoco nella stessa tipologia di oggetti*
   * *Dovrà essere presente una funzione che permette di eseguire la creazione dei mezzi di locomozione di una famiglia (N biciclette e 1 automobile), passando le quantità e i valori come parametri alla funzione CreaMezziDellaFamiglia(int numeroDiBiciclette, string string marcaBiciclette, string[] modelliBiciclette, string marcaAutomobile, string modelloAutomobile)*

[Tot: \_\_/10]