# Test di valutazione – Modulo 2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Filomena |
|  |  | Cognome | Iannone |
|  |  | 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*

*I pilastri della OOP sono l’incapsulamento, l’ereditarietà e il polimorfismo.*

*L’incapsulamento permette di racchiudere all’interno di una classe vari metodi che non saranno visibili nel momento in cui questa sarà richiamata. Permette all’utente di utilizzare le funzionalità nascondendo i dettagli dell’implementazione. L’ereditarietà è la proprietà per la quale è possibile creare nuove classi che abbiano proprietà di classi già esistenti estendendone le caratteristiche.*

*Il polimorfismo permette di utilizzare lo stesso codice con istanze di classi diverse.*

*Sono fondamentali per lo sviluppo orientato agli oggetti perchè permettono di far interagire gli oggetti e di ricavare molteplici proprietà grazie alla loro interazione.*

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*

*Per overloading si intende quando metodi e proprietà hanno lo stesso nome ma parametri diversi. Questa è un’operazione possibile perchè ciò che caratterizza un metodo o una proprietà è la firma (nome + numero e tipi di parametri).*

*Per override si intende la possibilità di sovrascrivere un metodo con un’implementazione definita successivamente.*

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

*Il Garbage Collector è un sistema di gestione automatico della memoria che permette di liberare aree di memoria che non vengono più utilizzate in modo tale da permettere ai processi che hanno bisogno di allocare più memoria di averla a disposizione quando serve.*

*In .NET permette di gestire la memoria allocata dagli oggetti per permettere a quest’ultimi di avere la memoria di cui necessitano a disposizione, quando serve.*

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

*Un Value Type (int, bool, double ecc) contiene al suo interno il dato che lo definisce e non è ammesso che accetti il valore nullo mentre il Reference Type contiene il riferimento ad un determinato oggetto e può assumere valore nullo nel momento in cui non referenzia nessuna instanza. Le classi sono Reference Type.*

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 sta per “Language Integrated Query” ed è un framework che mi permette di accedere ed interrogare delle sorgenti di dati all’interno del linguaggio. LINQ può interrogare qualunque tipo di dati che implementi Ienumerable.*

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 è un tipo di riferimento che permette di definire le cartteristiche di un insieme di oggetti che hanno le stesse proprietà e metodi.*

*Una classe è astratta quando contiene almeno un metodo astratto (un metodo che viene inizializzato e non implementato e al quale si può far riferimento per più di una classe).*

*Un’interfaccia definisce invece un contratto che deve essere rispettato dalla classe che la implementa; è priva di implementazioni e di modificatori di accessibilità. Si utilizzano le interfacce perchè in .NET l’ereditarietà multipla non è ammessa e qundi si ricorre alla proprietà delle classi di poter implementare più interfacce contemporaneamente.*

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

Una “classe parametrica” mi permette di definire una funzione generica che posso adattare a più funzioni, permette di scorporare la funzionalità dal contesto.

Esempio:

public abstract class ManagerBase <TEntity> dove TEntity indica il tipo

public class LibroManager : ManagerBase <Libri>

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]