# Test di valutazione – Modulo 2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Serena |
|  |  | Cognome | Curti |
|  |  | 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 fondamentali della OOP sono tre: incapsulamento, ederitarietà e polimorfismo.

Incapsulamento: la possibilità di accorpare metodi e e proprietà in un unico oggetto, così facendo si possono formare tante piccole parti.

Ereditarietà: la possibilità di far ereditare tutti i metodi e le proprietà da una classe ad una sottoclasse.

Polimorfismo: possibilità di una variabile di poter essere vista come tipologie diverse.

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*

Overloading: possono esistere metodi con lo stesso nome, l’importante è che la fima sia diversa ovvero il numero di parametri all’interno.

Overriding: Sovrascrivere un metodo per renderlo più generico.

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

Il Garbage Collector è un sistema di gestione della memoria ovvero gestisce l’allocazione e il rilascio. All’interno della memoria vengono marcati degli oggetti che non vengono più utilizzati, non vengono eliminati ma vengono appunto marcati come zombie. Se durante la compilazione di un programma non si ha più memoria, il Garbage Collector interviene eliminando questa lista di zombie per dare più memoria possibile. Viene gestito da .NET in maniera semi automatica ma è possibile fare una forzatura in situazioni particolari.

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

Una Values Type contiene un valore del tipo, per esempio una variabile *int* connterrà al suo interno un intero come 57 o 34, inoltre contengono sempre un valore, non è ammesso null. Una Reference Type non contiene direttamente un valore ma un riferimento ad 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 sta per Language Integrated Query è un framework che permette l’interrogazione di ogni sorgente di dati che rappresenti i dati sotto forma di oggetti.

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 componente che racchiude una serie di proprietà e metodi, può implementare al su interno più interfacce.

Una classe astratta verrà usata solo come classe base per altre classi, non è possibile creare un’istanza di una classe astratta. Le classi atratte contengono almeno un metodo astratto.

Un’interfaccia ha all’interno solo metodi e proprietà astratti, non deve essere specificato il modificatore di accessibilità. Definisce un contratto che la classe che la implementa deve per forza rispettare. Infatti l’interfaccia definisce solo la stesura del metodo, senza corpo, rendendolo quindi ancora più generico.

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

Si tratta della possibilità di creare delle classi con dei parametri di tipo, racchiusi nelle < >. Possono assumere diversi valori primitivi. Questo permettere di rendere il nostro codice ottimizzato.

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]