# Test di valutazione – Modulo 2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Francesca |
|  |  | Cognome | Bonin |
|  |  | Data | 18.10.19 |

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 sulla quale si basa la OOP sono il principio di ereditarietà e poliformismo. L’ereditarietà prevede l’esistenza di classi collegate tra di loro, questo consente di generalizzare e specializzare una classe. La classe derivata, ad un livello più basso, eredita le caratteristiche della classe base. Sintatticamente una relazione di derivazione si indica con classe derivata: classe base.*

*Il polimorfismo invece consente di creare forme diverse di un tipo base. Ad esempio posso creare una classe derivata e avere un metodo della classe base con lo stesso nome della classe derivata.*

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 significa creare un metodo o una funzione che ha lo stesso nome di un’altra ma passa parametri diversi, quindi ha una firma diversa.*

*Overriding invece significa creare due metodi che hanno la stessa firma, però uno sarà contenuto all’interno della classe base mentre l’altro sarà all’interno della classe derivata.*

*//Overloading*

 public int Somma(int a, int b)

    {

        int Somma = a + b;

    }

    public int Somma(int a, int b, int c)

    {

        int Somma = a + b + c;

    }

*//Overriding*

class Classebase {

    public virtual void Stampa()

    {

        Console.WriteLine("Hello world!");

    }

}

  class derivata : Classebase

{

    public override void Show()

    {

        Console.WriteLine("Helloo!");

    }

}

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

*È un processo di gestione automatica della memoria. Crea una lista di deletable elements ma non li cancella uno per uno perchè se no userebbe troppa memoria, quando la memoria è piena, per evitare di fare esplodere il sistema operativo, libera la memoria da questi deletable elements*

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

*I value type sono i tipi primitivi (Int, bool.. )mentre i Reference Type sono di tipo stringa array, classe. La differenza principale, oltre al tipo, è il modo in cui questi valori vengono allocati in memoria. Il primo tipo conserva il valore della variabile dentro il proprio memory space, mentre il secondo tipo non conserva il valore della variabile in maniera diretta, conserva invece un riferimento all’indirizzo di un’altra area di memoria che conserva il valore*

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, la sua importanza deriva dal fatto che consente di creare delle queries come su SQL ma all’interno del codice stesso, è appunto integrato all’interno del 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*

*Per quanto riguarda la differenza tra classe e la classe astratta, essa consiste nel fatto che la classe astratta non può creare oggetti, cioè istanze di quella classe, mentre la classe si.*

*La classe astratta ti crea un comportamento di default mentre l’interfaccia ti crea un vincolo. L’interfaccia per essere utilizzata ha bisogno di essere “chiamata” da una classe. Per quanto riguarda il livello di accesso ad un’interfaccia o ad una classe, l’interfaccia non ha l’indicazione pubblica o privata, sarà sempre pubblica*

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

Le classi parametriche hanno una o più variabili racchiuse all’interno della sintassi <>. Ad esempio definiamo la classe parametrica <Tentity> che racchiude al suo interno le entità Libro, Utente e Genere. Il fatto di utilizzare Tentiy consente di riutilizzare gli stessi pezzi di codice ma senza vincolare il codice ad un’entità specifica.

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]