# Test di valutazione – Modulo 2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Irene |
|  |  | Cognome | Musso |
|  |  | 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*

Sono ereditarietà, polimorfismo e incapsulamento. Il *polimorfismo* è la possibilità di trattare un’istanza di un tipo come se fosse un’istanza di un altro tipo. L’*ereditarietà* presenta una classe che deriva e che può usare tutti i membri della classe ereditata come se fossero propri.

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’overloading permette di dichiare metodi e proprietà con lo stesso nome, in quanto si utilizzano interfacce che possono predere facce (ossia ad esempio classi) diverse. L’overriding permette una nuova implementazione di un membro ereditato da una classe base.

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

Il Garbage Collector permette di gestire la memoria in modo automatico utilizzando il Costruttore e il Distruttore. Elimina gli “Zombie”, le porzioni di memoria non più utilizzate, e permette di sviluppare l’applicazione senza preoccuparsi di liberare la memoria per gli oggetti creati.

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

* I Value Type sono ad esempio int, bool, double; contengono direttamente il dato e copiandoli si copia il dato in esso contenuto. Le modifiche hanno effetto solo sull’istanza corrente.
* I Reference Type sono ad esempio tutte le classi; contengono solo il riferimento ad un oggetto e copiandoli si copia solo il reference. Le moFdifiche hanno effetto sull’oggetto a cui puntano.

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 permette la scrittura di query con una sintassi simile a SQL direttamente in linguaggio C#, è il nome dato all’insieme di comandi che permettono ciò. È diventato fondamentale perchè permette il filtraggio e l’ordinamento dei dati in modo veloce; inoltre è facilmente gestibile ed immediato.

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

La classe è definita e contienue caratterstiche specifiche e può essere istanzializzata. La classe astratta non può essere istanzializzata e ha una base reale. L’Interfaccia è solo una dichiarazione di contratto, un vincolo, una classe astratta che non ha alcun metodo implementato. La classe astratta contiene almeno un metodo astratto, ossia che dichiara solo la sua firma senza implementazione. L’interfaccia a differenza delle altre non ha modificatore di accessibilità. Una classe può 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

È una classe che si basa su elementi generics.

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]