# Test di valutazione – Modulo 2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Katherine |
|  |  | Cognome | Gigremosa |
|  |  | 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 sulla quale si basa la OOP sono: incapsulamento, ereditarietà e polimorfismo.*

*Incapsulamento: è l’insieme di proprietà e metodi e vengono considerati come una singola entità*

*Ereditarietà: significa creare delle sottoclassi che ereditano da una classe base esistente creando un legame gerarchico dove nella classe base ci sono gli stati e i comportamenti comuni a tutte le sottoclassi, mentre ogni sottoclasse ha la sua specializzazione*

*Polimorfismo: la capacità di gestire più classi e di utilizzarne i metodi e le proprietà in modo indipendente tra loro a seconda della classe che si sta utilizzando anche se hanno lo stesso nome CREA, CANCELLA ecc..*

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*

*Si parla di overloading quando esistono metodi e proprietà con lo stesso nome, non fanno però riferimento alla stessa cosa perchè possiedono firme differenti*

*Es. Il metodo della somma di 2 numeri sarà diversa dalla somma di 3 numeri*

*public int somma(int a, int b, int c) != public int somma(int a, int b, int c)*

*L’overriding invece è quando il metodo della sottoclasse prevale sulla classe base che si differenzierà completamente nelle funzionalità*

*Es. Avente una classe base Veicolo e la sottoclasse Auto, avranno entrambe il metodo Accellera, ma nel momento in cui stiamo utilizzando Auto utilizzeremo l’Accellera di Auto che sarà diversa nel codice all’Accelera di Veicolo che sarà generale*

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

*Il Garbage Collector è una funzionalità per la gestione automatica della memoria, libera porzioni di memoria non più utilizzate dall’applicazione ogni tot di tempo per non riempire la memoria e rallentare i processi*

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

*I Values Type salvano nello spazio di memoria il valore, i tipi primitivi (int, double, bool...) sono di tipo Value*

*Quando si fa un assegnazione il valore all’interno della variabile viene copiata e ci si potranno fare delle operazioni distinte*

*I Reference Type hanno nello spazio di memoria l’indirizzo di dove si trova il valore e possono essere gli array, le stringhe ecc..*

*A differenza dei Values type, ai Reference Type non viene copiato il valore ma punterà allo spazio di memoria di dove si trova quell’istanza*

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 è l’acronimo di Language Integrate Query è molto importante perchè ti consente di scrivere query all’interno del linguaggio .Net per recuperare i dati dalle diverse risorse*

*Es. UPDATE,, CREATE..*

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

*L’interfaccia definisce ciò che deve fare una classe senza fornire i dettagli, esempio: posso definire un’ interfaccia FiguraGeometrica che contiene i metodi GetArea e GetPerimetro, visto che le classi possono essere triangoli, cerchi, quadrati ecc.. oguno avrà dei modi diversi per calcolare area e perimetro. E questi vanno definiti nella classe*

*Una classe astratta è simile a un interfaccia ma invece che dire solo che la classe deve avere i 2 metodi GetArea e GetPerimetro, fornisce anche delle implementazioni che puoi utilizzare o meno ed effetuare un override, infatti generalmente in una classe abstract i metodi sono definiti con virtual*

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 sono classi generiche dove cioè non indichi i tipi utilizzati internamente ma crei un’astrazione, per esempio se tu volessi creare un oggetto lista che può contenere oggetti di qualsiasi tipo, un esempio di definizione della classe può essere class List<T>{} dove poi l’oggetto T è quello generico

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]