# Test di valutazione – Modulo 1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Filomena |
|  |  | Cognome | Iannone |
|  |  | Data | 11/10/2019 |

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. *Quali tra i seguenti è un IDE? Per quelli che non sono IDE, spiegare per quale motivo non è ritenuto tale*

Notepad – è un editor di testo senza supporto, a differenza dell’IDE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Visual Studio Code – è un editor di codice sorgente\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

.NET – è un framework che permette di sviluppare più linguaggi\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Visual Studio - \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[CORREZIONE] VS Code è considerato un IDE per via dei plugin

1. Cos’è una funzione?

Un blocco di codice che esegue determinate istruzioni

È un’eccezione non gestita

È una lista di elementi

È una struct con determinati livelli di accesso

1. *Cosa sono le proprietà, le costanti e le variabili? Dare una descrizione dettagliata di ciascun elemento.*

*Nella programmazioe ad oggetti (OOP) le proprietà o attributi di una classe sono le caratteristiche specifiche che identificano un oggetto. Insieme ai metodi, che permettono di manipolare i valori in uscita, caratterizzano il comportamento di una classe.*

*Le variabili sono contenitori di dati destinati a contenere valori che possono essere modificati durante l’esecuzione del programma.*

*Al contrario le costanti contengono valori che non possono essere modificati.*

1. *Cosa sono gli operatori logici e a cosa servono? Fornire una serie di esempi di utilizzo in codice C#.*

*Gli operatori logici permettono di mettere in relazione tra di loro due operandi di tipo booleano e che quindi restituiscono una valore di tipo “vero” o “falso”.*

*Esempi dal codice al punto 10:*

valInt >= minVal && valInt <= maxVal : operatore and, verifica che entrambe le condizioni siano verificate;

isInt == false || InRange == false : operatore or, verifica che o una o l’altra condizione siano verificate.

1. *Elencare quante più differenze possibili da .NET Core e .NET Framework?*

*.NET Core e .NET framework sono entrambi implementazioni di .NET.*

*.NET Framework è la prima versione e funziona solo su Windows mentre .NET Core è più recente ed è multipiattaforma. E’ infatti operativo su Windows, Linux, MacOs.*

*Il codice è quasi sempre portabile tra le 2 implementazioni a patto che si rispettino le tabelle di compatibilità.*

*Entrambe si basano sui contratti scritti nella .NET Standard Library e hanno un’infrastruttura comune costituita dai compilatori, dai linguaggi e dal RunTime.*

1. *Qual è la differenza tra una proprietà, un metodo o un campo “private” e “public” in una classe?*

*Una classe è caratterizzata da proprietà, metodi e campi. Le proprietà o attributi di una classe sono le caratteristiche specifiche che identificano un oggetto (istanza di una classe). Insieme ai metodi, che permettono di manipolare i valori in uscita, caratterizzano il comportamento di una classe.*

*Definire una classe come “private” o “public” permette di stabilire la visibilità della stessa all’interno del codice. In particolare con “private” la visibiità è privata e la classe non può essere vista e richiamata all’esterno dell’istruzione alla quale appartiene. Al contrario “public” permette di utilizzare la classe anche all’esterno dell’istruzione nella quale è definita.*

1. *Quali sono le differenze tra Do-While, While, For, For-Each e Switch? Fornire una descrizione ed un esempio di ciascuno di essi*

*Do-While mi fa eseguire una determinata istruzione finchè la condizione nel while rimane valida.*

*Do {...} while (i <= 10)*

*While permette di eseguire l’istruzione successiva finchè la condizione dichiarata al suo interno sussiste.*

*While (i <=10) {...}*

*For permette di inizializzare un indice, di impostare una condizione che deve essere verificata e di impostare un’operazione che permette l’incremento dell’indice.*

*For (i = 0; i <=10, i++) {...}*

*For-Each permette di scorrere gli elementi di un array e di eseguire l’istruzione contenuta al suo interno per ogni elemento*

Foreach (person in rubrica) {...}

Switch permette di scegliere tra un determinato numero di opzioni e di eseguire l’istruzione relativa

Switch (selezione) {

Case 1: ... ;

break;

Case 2: ... ;

break; }

1. *Cosa sono le eccezioni, a cosa servono e quando non dovrebbero essere mai utilizzate?*

*Le eccezioni sono situazioni impreviste che possono verificarsi durante l’esecuzione di un programma. Servono a prevenire eventuali errori di esecuzione e a far si che il programma riconosca l’errore e continui a funzionare anche nel caso in cui uno di questi si verifichi.*

*Ad esempio potrebbe capitare che si scriva un’istruzione che mi deve restituire il risultato della divisione di 2 numeri e che, per qualche motivo, questo risultato non sia ottenibile. Per impostare l’eccezione si utilizza il costrutto try...catch che permette di far eseguire l’istruzione di divisione finchè non occorra un problema per il quale questo risultato non sia ottenibile. In quest’ultima evenienza si imposta nel catch l’istruzione che mi segnala l’esistenza dell’eccezione restituendomi un valore di default e permettendomi di proseguire con l’esecuzione del programma. Le eccezioni non dovrebbero essere utilizzate quando si è in presenza di un errore causato da un’errata digitazione da console. Questo tipo di errori sarebbe meglio gestirli richiedendo all’utente di reinserire i dati e segnalando l’errore in modo tale da prevenirlo all’occasione successiva.*

1. *Qual è la differenza tra un Array e una Lista? Fornire un esempio di ciascuno di essi, come si esegue la creazione e la popolazione di queste strutture.*

*Gli array e le liste sono entrambi contenitori di dati. Mentre l’array può contenere una sola tipologia di dati, la lista può archiviare anche tipi di dati diversi.*

*Per gli array è necessario conoscere preventivamente il numero di elementi da inserire mentre la lista permette di inserire elementi fino al termine dell’inserimento stesso (arbitrario). Gli array pemettono di allocare una memoria inferiore rispetto alle liste e sono preferibili nel caso di un numero di elementi minore.*

*Esempio creazione e popolazione array di tipo stringa:*

string[] sourceArray = new string[3]; //Creo l’array

for (var i = 0; i < 3; i++) //Scorro l’array inserendo un valore per ogni posizione.

{ var valore = Console.ReadLine();

sourceArray[i] = valore; }

*Esempio creazione e popolazione array di tipo stringa:*

List<Person> persone = new List<Person>(); //Creo la lista

foreach (var currentRow in tutteLeRigheDelDatabase) //scorro gli elementi dell’array

//creo l’oggetto persona

persone.Add(currentPerson); //aggiungo gli oggetti persona alla lista

[CORREZIONE]: Le lista non possono contenere valori di tipo diverso

1. *Esercitazione:*

*Si chiede di creare un app console in .NET Framework Core 3.0 che in fase di esecuzione richieda l’inserimento di un numero compreso da 1 e 10; dopo la lettura del numero, dovranno essere creati un pari numero di oggetti “prodotto” (classe Product), ciascuno dei quali caratterizzato da codice (alfanumerico, campo “Code”) e nome (campo “Name”). Una volta terminato l’inserimento dei prodotti, gli stessi devono essere stampati a video e scritti all’interno di un file di testo in modo tale da poter essere facilmente riletti (la funzione di rilettura non è richiesta).*

*[CORRETTO]: Programma funzionante, ma il codice non è formattato adeguatamente. Questa cosa, a tendere, comporta una poca manutenibilità del codice e difficoltà di comprensione dello stesso*

[Tot: 8/10]