# Test di valutazione – Modulo 1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Francesca |
|  |  | Cognome | Bonin |
|  |  | Data | 11.11.19 |

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. *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 per cui ha meno funzionalità e supporto rispetto a un IDE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

.NET - \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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.*

La variabile può essere pensata come un contenitore di dati con un valore assegnato, il valore può cambiare nel corso dello sviluppo di un programma. A differenza di una variabile la costante non può modificare il suo valore, nel momento in cui ad essa viene assegnato un valore, esso non cambierà mai. Le costanti si indicano con const.

La proprietà contiene al suo interno variabili e/o metodi, essa accetta un metodo get che legge il valore e set che imposta il valore. L’accesso ad essa dipenderà dalla diversa assegnazione delle keyword private e public

a= 10; // variabile

const b=11; //costante

public class Persona // proprietà

{

Public string Nome { get; set; }

}

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

Gli operatori logici creano una condizione attraverso un sistema di verifica. Ne esistono di 3 tipi: AND, OR, NOT. In c# vengono indicati con &&, || e ! rispettivamente. Questi sono alcuni esempi, prendendo 2 ipotetiche variabili a e b:

if (a == 0 && b== 0)

{

Console.WriteLine(“entrambe le variabili sono pari a 0”)

}

if (a == 0 || b== 0)

{

Console.WriteLine(“Bravo!”)

}

if (a != 0)

{

Console.WriteLine(“Il valore è valido”)

}

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

.Net ha 2 framework principalmente: Core e Framework. Essi presentano delle differenze:

* Core è più flessibile rispetto a Framework in quanto permette di utilizzare più versioni sulla stessa macchina (e.g. versione 2.1 e 3.0 insieme). Framework può utilizzare solo la versione installata sulla macchina
* Core è cross-platform per cui funzionerà su tutti i sistemi operativi, a differenza della versione precente Framework che funziona solo su Windows

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

Private e public sono riferiti all’ accessibilità di una proprietà, un metodo o un campo all’interno di una classe. Se l’elemento in questione è pubblico sarà accessibile da qualsiasi punto all’interno del codice. Nel caso contrario, se fosse private non potrei accedervi nel momento in cui esco dalla classe che lo contiene. Solitamente i metodi vengono dichiarati come pubblici mentre i campi vengono definiti come privati.

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

La differenza principale tra il while e il Do while è che nel secondo caso, prima viene eseguita un’azione e poi viene controllata la condizione. Per cui anche se la condizione è falsa, l’azione verrà eseguita almeno una volta.

Il ciclo for, data una determinata condizione, ripete un’azione fino al termine del ciclo. A meno che il ciclo non venga interrotto ad un certo punto dalla keyword break.

La differenza principale tra for e foreach è che nel secondo caso non c’è bisogno di indicare un indice. Si tratta di un metodo più coinciso.

Switch consente la selezione di determinate alternative definite con la keyword case. Si tratta di un’alternativa all’utilizzo di if else. Rende il codice più leggibile.

Ipotiziamo di avere un while statement che funziona finchè l’indice di una lista di partecipanti è minore della sua lunghezza:

int index = 0;

while (index < Numero\_Partecipanti.Length)

{

Console.WriteLine(Numero\_Partecipanti[index].Nome + ", " + Numero\_Partecipanti[index].Cognome);

index = index + 1;

}

For e foreach:

for (int i = 0; i < Numero\_Partecipanti.Length; i++)

{

Console.WriteLine(Numero\_Partecipanti[i].Nome + ", " + Numero\_Partecipanti[i].Cognome);

}

foreach (Person current in Numero\_Partecipanti)

{

Console.WriteLine(current.Nome + ", " + current.Cognome);

}

Ipotizzando di avere una classe Animal:

switch ()

{

case Animal.Mammifero:

Console.WriteLine("L’animale scelto è un mammifero");

break;

case Animal.Rettile:

Console.WriteLine("L’animale scelto è un rettile");

break;

case Animal.Anfibio:

Console.WriteLine("L’animale scelto è un anfibio");

break;

default:

Console.WriteLine("L’animale selezionato non è appartenente a nessuna delle 3 classi");

break;

}

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

Le eccezioni sono un costrutto che viene utilizzato come barriera difensiva per evitare che il programma crashi nel momento in cui si verifica qualcosa che non era stato previsto nel codice.

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.*

*La differenza principale tra una lista e un array è che, il numero di elementi contenuti all’interno dell’array è prestabilito. Questo non accade nella lista, dove gli elementi al suo interno possono essere rimossi o aggiunti successivamente alla sua creazione.*

List<Product> prodotti = new List<Product>();

int[] targetArray = new int [6];

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).*

[Tot: \_\_/10]