# Test di valutazione – Modulo 1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Deborah |
|  |  | Cognome | Tucci |
|  |  | Data | 05/02/2021 |

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. *Spiegare le differenze tra memoria principale, secondaria e virtuale*

La memoria principale è quella memoria nella quale vengono caricati il programma e i dati annessi per eserguirlo nel breve termine, infatti mantiene l’informazione solo per il tempo dell’esecuzione. La memoria principale si differenzia dalle altre per la sua volatilità, infatti quando la macchina viene spenta perde i dati contenuti al suo interno; per la sua velocità nel scrivere e recuperare i dati; infine, per il suo accesso random, cioè il tempo di accesso alle celle è sempre lo stesso per ogni cella a prescindere dalla posizione. Un esempio è la RAM e la Cache.

Invece, nella memoria secondaria vengono memorizzati tutti i programmi e i dati annessi a lungo termine, in maniera permanente. Inoltre si differenzia dalle altre per la sua non volatilità, cioè anche quando viene spenta la macchina le informazioni contenute non vengono perse. Infine, è caratterizzata dall’accesso sequenziale, cioè il tempo per accedere a una cella dipende dalla sua posizione, e questo rallenta la sua velocità. Un esempio è il Disco Rigido, una pen-drive o un CD.

Mentre, mentre la memoria virtuale è quel meccanismo del sistema operativo per cui per un’applicazione da eseguire la capacità della memoria principale sembra maggiore rispetto a quella effettiva. Questo perchè quando l’applicazione deve essere eseguita nella memoria principale vengono caricati sono le istruzioni che devono essere eseguite nell’immediato, lasciando il resto nella memoria secondaria. Le altre istruzioni verranno poi chiamate nella memoria principale mediante il meccanisco di swap, che le sostituirà con le istruzioni già eseguite e non più necessarie.

1. *Cosa significa che un linguaggio è tipizzato? Spiegare la suddivisione in tipi in C# e le relative caratteristiche.*

C# è un linguaggio tipizzato in quanto garantisce che il valore di una variabile sia usato coerentemente in base al tipo della variabile. Un tipo infatti descrive l’identità della variabile e ne caretterizza le funzionalità ammesse. Esempio il tipo int ammette solo valori interi e permette di eseguere delle operazioni aritmetiche, a differena del tipo string.

In C# principalmente di distinguono i Value Type e i Reference Type.

I value type comprendono i tipi primitivi come int, bool, ma anche gli enum. Essi contengono sempre un valore e il dato è contenuto nello stack.

Una copia di un value type comporta la copia dei dati in esso contenuti.

Mentre, i reference type contengono solo un riferimento ad un oggetto, quindi la copia di un reference typer comporta la duplicazione della referenza. Essi sono contenuti nell’heap. Un esempio di reference type è la classe.

1. *Cos’è un sistema operativo? Dare una spiegazione delle funzionalità.*

Il sistema operativo è un insieme di software utile per l’accesso alle risorse del computer. Le sue funzionalità sono:

* gestione del processore, mediante il Kernel, garantendo il multitasking e l’esecuzione sicura dei programmi.
* gestione dell’interfaccia utente, con il FileSystem, che mostra e informazioni contenute nella memoria, mediante struttura gerarchica di cartelle e file.
* gestione della memoria, coordinando le memorie della macchina e assegnando parti di essa ai programmi da eseguire.
* gestione I/O, mediante il buffer, rendendo omogenea l’interazione dell’utente con la macchina, attraverso le diverse periferiche che hanno velocità e caratteristiche diverse.

1. *Descrivere il funzionamento degli operatori logici e delle relative tabelle di verità.*

Gli operatori logici, soprattutto quelli di uguaglianza, condizionali e di relazione sono utilizzati per definire le condizioni negli statement condizionali e iterativi.

In particolare l’operatore condizionale && (And), che confronta almeno due condizioni, in base alla relativa tabella di verità, risulta sempre falsa, tranne nel caso che tutte le condizione concatenate siano vere.

Invece, con l’operatore || (OR) , in base alla relativa tabella di verità, la condizione risulta sempre vera, tranne nel caso che tutte le condizione concatenate siano false.

1. *Spiegare la differenza tra ricorsione e iterazione e le relative peculiarità.*

La ricorsione si ha quando un’istruzione di una funzione richiama la funzione stessa; questa operazione occupa più spazio in memoria. Mentre, con l’iterazione si esegue un’istruzione finchè non viene soddisfatta una determinata condizione.

Solitamente la ricorsione è meno performante dell’iterazione.

1. *Dare una definizione di casting. Fornire degli esempi.*

Si definisce casting l’operazione di conversione di un tipo in un altro tipo affine. Non sempre i tipi sono compatibili e quindi si genera una eccezione.

Si può eseguire un cast esplicito, come ad esempio

double x = 2;

int y = (int) x;

oppure il cast implicito; o in certi casi bisogna avvalersi dei costrutti TryParse o Convert, ad esempio

double x = 2;

int y = Convert.ToInt32(x);

1. *Spiegare cosa è una routine. Specificare le tipologie di routine e fornire degli esempi per ciascuna.*

Una routine è un insieme di istruzioni contenute in un blocco di codice e che può essere richiamata in un altro punto di codice. Essa può contenere dei parametri di inputi e/o un valore di ritorno il cui tipo è definito nella firma della routine.

Vi sono due tipologie di routine: le funzioni e le procedure.

Le funzioni, a differenza delle procedure, hanno il valore di ritorno, mentre le procedure non hanno un return e sono definite con void.

Esempio di procedura:

public static void ViewNum(int num)

{

If(num != 0)

Console.Write("{0} ", num);

}

Esempio di funzione:

public static int Num(int num)

{

If(num != 0)

return num;

}

1. *Descrivere la funzione del compilatore*

Il compilatore è quello strumento che si occupa di tradurre il codice sorgente, che è in un linguaggio di programmazione, in codice eseguibile in linguaggio macchina, controllando la sintassi del linguaggio di programmazione.

1. *Descrivere le funzioni di un sistema di versionamento e le diverse tipologie*

Un sistema di versionamento è un sistema di controllo delle versioni di un progetto, registrando nel tempo i cambiamenti ad uno o più file di esso. Questo consente di ripristinare i file o il progetto ad una versione precedente e tener traccia delle modifiche apportate nel tempo.

Vi sono diverse tipologie di sistema di controllo versione:

* Locale, che salva in locale l’insieme delle versioni.
* Centralizzato, caratterizzato da un unico server esterno che memorizza le varie versioni dei file e gli utenti li scaricano direttamente da questo server centrale.
* Distribuito, caratterizzato da un repository, che permette ai client di controllare la versione più recente del file e tener traccia dello storico delle modifiche apportate.

*Esercizio Pratico*

Creare una Console Application che gestisca i Task dell’utente.

Per Task viene inteso un oggetto che ha una descrizione, una data di scadenza e un livello di importanza (Basso, Medio, Alto).

L’utente può:

* Vedere i Task inseriti
* Aggiungere un nuovo Task
* Eliminare un Task
* Filtrare i Task per importanza

Requisiti Tecnici:

-Salvare i Task in un file

-Utilizzare adeguatamente il concetto di classe

-Dividere le funzionalità in relative funzioni e procedure

-Commentare

-Mettere una nomenclatura conforme

-Le date di scadenza devono essere posteriori o uguali rispetto alla data di inserimento

-Controllare l’input utente

Opzionale: Utilizzare Enum

Mettere il codice dell’esercizio in un Repository di GitHub.