# Test di valutazione – Modulo 1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Elena |
|  |  | Cognome | Zazzetti |
|  |  | Data | 05/02/2020 |

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*

*Nella memoria principale, o centrale, vengono caricati i programmi e i dati necessari per eseguirlo. Si definisce di breve-medio termine poiché l’informazione è mantenuta per il solo tempo di esecuzione del programma. È inoltre volatile, cioè l’informazione è persa se la macchina viene spenta. Le informazioni sono salvate in celle di memoria <indirizzo,contenuto> e l’accesso avviene in modo random, non vi è quindi una differenza nel tempo di accesso alle varie celle.*

*La memoria secondaria, a differenza della principale, immagazzina il programma e i dati in memoria permanente. Si definisce dunque di lungo termine. Questa non è volatile in quanto le informazioni non vengono perse quando la macchina viene spenta. L’accesso alle celle di memoria è sequenziale, quindi dipendono dalla posizione delle celle. La capacità è tendenzialmente maggiore di quella della principale mentre la velocità è più limitata.*

*La memoria virtuale è invece un meccanismo per cui la RAM (memoria principale) risulta essere di capacità maggiore rispetto alla sua capacità fisica effettiva. Questo avviene poiché per eseguire un programma le istruzioni relative che non servono nell’immediato vengono salvate nella memoria secondaria. Attraverso uno swap tra memoria principale e secondaria nelle principale vengono salvate le informazioni che devono essere eseguite nell’immediato. A questo punto il processore eseguirà le istruzioni presenti in memoria principale, poiché grazie al principio di località eseguirà le istruzioni allocate in zone di memoria più vicine. Con questo procedimento è possibile quindi eseguire applicazioni che richiedono una memoria maggiore rispetto alla memoria effettiva della RAM.*

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

*Nel linguaggio tipizzato le variabili vengono dichiarate di un certo tipo, che ne descrive il ruolo e le funzionalità e permette al linguaggio di controllare che i valori che la variabile assuma siano coerenti con il tipo definito.*

*Si differenziano in Value Type e Refernce Type.*

*I primi contengono direttamente il valore dei dati, salvato nello stack, e devono contenere sempre un valore(non possono essere null). Inoltre una copia di un value type comporta la copia dei valori contenuti. I value type in c# sono: int, byte bool double float ... , Mentre sono value type anche i tipi definiti dall’utente enum e struct.*

*I reference Type contengono invece il riferimento alla locazione di memoria della variabile. A differeza dei Value type, i reference type vengono gestiti nell’heap. Questi possono essere nulli, cioè reference che non referenziano nessuna istanza. I reference type in C# sono le classi e le stringhe.*

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

*Un sistema operativo è un insieme di softwares che permettono di sfruttare tutte le risorse della macchina. Le sue funzionalità riguardano:*

* *La gestione del processore attraverso il Kernel, il nucleo del sistema operativo, che gestisce le funzionalità del processore permettendo un esecuzione sicura dei programmi, facendo in modo che ogni applicazione non vada a sovrascrivere spazi di memoria destinate ad altre, e l’esecuzione di più programmi in parallelo.*
* *La gestione I/O. Le operazioni di input/ouput vengono ricevute tramite periferiche. L’interazione dell’utente con le diverse periferiche viene resa omogenea attraverso l’uso del buffer, area di memoria destinata all’accumulo temporaneo delle informazioni delle periferiche, che eguaglia i tempi di esecuzione di ciascuna periferica.*
* *Gestione della memoria. Il sistema operativo si occupa di gestire la memoria assegnata a ciascun programma di esecuzione e coordinare le azioni tra le memorie della macchina (come spiegato al punto 1).*
* *Gestione interfaccia utente. Questa comprende bash, interfaccia grafica tramite cartelle, icone ecc, e interfaccia vocale.*

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

* *AND*

***A B A AND B***

*F F F*

*F V F*

*V F F*

*V V V*

* *OR*

***A B A AND B***

*F F F*

*F V V*

*V F V*

*V V V*

* *NOT*

***A*** ***NOT A***

*F V*

*V F*

* *XOR*

***A B A AND B***

*F F F*

*F V V*

*V F V*

*V V F*

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

*L’iterazione svolge ripetutamente un blocco di codice fino a che una condizione specificata non venga soddisfatta. La ricorsione è uno statement di una funzione che richiama la funzione stessa. L’iterazione è più efficiente in termini di memoria mentre in termini di tempo varia a seconda del numero delle iterazioni / richiami della funzioni.*

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

*Il casting è una conversione di un tipo ad un tipo affine.*

*Questo può essere implicito, tendenzialmente quando il range di valori del tipo da convertire è contenuto nel range del tipo in cui lo si voglia convertire. Non c’è perdita di dati con questo tipo di conversione.*

*Es. byte b=10; int a=b;*

*Il casting può essere esplicito mediante l’uso di parentesi tonde o la keyword as:*

*Es. double b;*

*int a= (int) b;*

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

*Una routine è un insieme di istruzioni racchiuse in un blocco di codice e richiamabile.*

*La routine viene dichiarata per mezzo di: keyword (zero o più), return type, nome della routine, parametri in input.*

*Si definiscono routine le funzioni, che hanno un valore di ritorno, e le procedure che non hanno valore di ritorno (void).*

*Es.*

*int Sum( int x, int y){ // funzione*

*return x+y;}*

*void print(int x){ //procedura*

*Console.WriteLine(x);}*

1. *Descrivere la funzione del compilatore*

*Il compilatore effettua un controllo sulla sintassi del linguaggio di programmazione (non la logica) e traduce il codice sorgente in codice macchina.*

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

*Un sistema di versionamento permette di registrare lo storico dei cambiamenti di un progetto e file relativi con la possibilità di recuperare le versioni precedenti del progetto o di file specifici.*

*Il primo sistema di controllo di versione era quello locale. Questo salvava in locale su disco il progetto e i file relativi e in questo modo non c’era la possibilità di lavorare in condivisione allo stesso progetto. Questo sistema di controllo fu abbandonato per il centralizzato.*

*Il sistema di controllo di versione centralizzato è costituito da un unico server esterno che contiene le versioni del progetto, con la possibilità di avere più utenti che connessi al server possono scambiare file con questo. Inoltre un amministratore che gestisce il server può decidere l’accesso ai file per i vari utenti. Uno svantaggio è il downtime.*

*Sistemi di controllo distribuiti servono per controllare lo snapchat più recente del file e avere tutto lo storico delle modifiche della repository. Più utenti vi possono collaborare, caricare la loro versione della repository e scaricare quella attuale.*

*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.