# Test di fine settimana – Week2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Laura |
|  |  | Cognome | Pennetta |
|  |  | Data | 28/05/2021 |

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. *Dare una definizione di classe statica*

*Una classe è statica quando tutti i suoi membri sono statici.*

*Non è possibile crearne istanze.*

*Una classe statica può essere usata come contenitore di metodi con solo paramentri di input e output.*

*Una classe statica non può essere ereditata.*

1. *Quali tipologie di ereditarietà sono consentite in C# e come si definisce? Fornire un esempio*

*I principi fondamentali della programmazione orientata agli oggetti sono tre: INCAPSULAMENTO, EREDITARIETA’,POLIMORFISMO. L’eriditarietà si basa su una relazione gerarchica tra diverse classi. Si ha quindi la classe base e la sottoclasse anche detta classe derivata; tutte le classi derivano implicitamente dalla classe Object. Esistono diversi tipi di ereditarietà:*

* *Single: abbiama una classe derivata ed una base;*
* *Gerarchica: abbiamo più classi derivate ed una base;*
* *Multipla: abbiamo una sottoclasse derivata da più classi base (non possibile con C#);*
* *Multilivello:abbiamo più classi, ognuna derivante da un’altra, quindi, si ha un ereditarietà a più livelli.*

1. *Elencare le principali caratteristiche della classe System.Object.*

*È la principale classe base.*

*Tutto ciò che deriva da Object ne eredita i metodi, o meglio, le classi derivate possono eseguire l’ovveride di alcuni metodi:*

* *ToString:converte l’oggetto in una stringa;*
* *GetHashCode:ottiene il codice hash di un oggetto;*
* *Equals:permette la comparazione fra oggetti;*
* *Finalize:chiamato in fase di cancellazione dal garbage collector;*
* *GetType:ottiene il tipo dell’oggetto;*
* *Memberwise Clone:effettua la copia dell’oggetto e ritorna una reference della copia.*

*Se specific overraid davanti ad uno dei metodi di Object, allora, posso sovrascrivere il metodo.*

1. *Descrivere le due fasi di gestione delle eccezioni.*

*La gestione delle eccezioni è fondamentale nell’interazione con l’utente, infatti, gestico in modo specifico o generale gli eventuali errori che l’utente potrebbe commettere. Per la gestione degli errori abbiamo tre termini fondamentali:*

* *Try: racchiude gli statement per i quali si possono avere errori;*
* *Catch: cattura l’errore;*
* *Finally: serve per chiudere la connessione.*

*È preferibile trattare le eccezioni in modo specific ma, qualora non si riuscisse, andrebbe bene trattarle in modo generico tramite l’ausilio del termine Exception (eccezione più generica).*

*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:

- Recuperare i task da file

- Salvare i Task in un file

- Utilizzare adeguatamente il concetto di classe

- Dividere le funzionalità in relative funzioni e procedure

- Commentare le scelte algoritmiche

- Mettere una nomenclatura conforme

- Verificare le date di scadenza dei task: 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.