# Test di fine settimana – Week2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Eleonora |
|  |  | Cognome | Lombardo |
|  |  | 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 statica è una classe definita con la parola chiave static ed ha tutti i membri statici, di questa classe non si possono creare istanze e quindi non si può usare l’operatore new, si accede ai membri della classe con il nome della classe stessa, non possono essere ereditate e non contengono un costruttore.*

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

*In c# è possibile ereditare da una classe, ereditare “a catena” e avere più classi derivate ma non è possibile ereditare da più classi in contemporanea, cioè non si possono avere cose del tipo:*

*Madre padre*

*figlio*

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

*La classe Object è la classe che sta all’apice della gerarchia delle classi, la “radice”.  
Grazie ad essa è possibile usare le classi ed è formata da i seguenti metodi virtuali:*

*Equals, ReferenceEquals, GetHashCode, GetType, ToString, RememberWIseClone.*

*I metodi più utilizzati sono Equals per vedere se due oggetti sono uguali, GetHashCode che restituisce l’hash del’oggetto, GetType che ritorna il tipo dell’oggetto in questione e ToString che “casta” a stringa i “metadati” dell’oggetto.*

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

*Intercettare e agire che vengono fatte con try e catch (e finally)*

*Nel blocco di codice del try eseguiamo l’istruzione “a rischio errore” così nel caso in cui ve ne sia uno può essere intercettato dal catch, blocco in cui si gestisce l’eccezione.*

*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.