# Test di fine settimana – Week2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Mariangela |
|  |  | Cognome | Leone |
|  |  | 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 contiene solo membri statici, ovvero elementi associati direttamente al tipo e condivisi con tutte le istanze; per rappresentarle si usa la keyword static.*

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

*L’ereditarietà è il secondo principio cardine della programmazione ad oggetti basato sulla relazione gerarchica tra classi diverse. Ci sono tre tipi di ereditarietà: semplice, definisce legame diretto tra sottoclasse (classe derivata, Square) e superclasse (classe base, Shape); gerarchica, in cui sono presenti più square; multilivello, in cui l’eredietarietà avviene in modo indiretto.*

*Eesempio: class Studente : Persona (studente pè la sottoclasse, persona è la superclasse)*

*{*

*Public String Matricola {get; set; }*

*Matricola aggiunge una caratteristica alla classe Persona.*

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

*-ToString: converte l’oggetto in una stringa;*

*-GetHashCode: ottiene il codice hash della classe;*

*-Equals: permette di fare la comparazione tra due oggtti mediante la scelta di una caratteristica;*

*-Finalize: viene chiamato in fase di cancellazione dal Garbage Collection perchè le classi sono Reference Type quindi nello heap;*

*-GetType: ottiene il tipo dell’oggetto;ù*

*-MemberwiseClone: effettua la copia, ovvero la copia dell’indirizzo, otenendo una copia del riferimento.*

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

*Tramite il Try si racchiudono gli statement per i quali si vogliono trovare gli errori; tramite il catch viene catturato l’errore specifico; finally serve per indicare il comando finale da eseguire in ogni caso.*

*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.