# Test di fine settimana – Week2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome |  |
|  |  | Cognome |  |
|  |  | Data |  |

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*

**Si tratta di una classe che con proprietà e metodi statici, che non possono essere generalizzati o specializzati. Definiscono le caratteristiche di quella classe specifica.**

**Es: la classe relativa ad un gioco specifico. Le regole del gioco sono specifiche del gioco stesso e quindi “proprietarie” di quella spacifica classe.**

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

**L’ereditarietà è uno dei principi della OOP. Si base sull’acquisire le proprietà della classe padre da parte delle classi figlie. Questa possibilità permette di creare delle sotto-classi (SO-C) che utilizzano il comportamento della super-classe (S-C), eventualmente specializzandolo ulteriormente. In C# abbiamo ereditarietà:**

* **Semplice: figura geometrica (S-C), triangolo (SO-C)**
* **Gerarchica (dà luogo al polimorfismo): figura geometrica (S-C), triangolo (SO-C)/quadrato (SO-C)**
* **Multipla (non usata in C#): la SO-C eredita proprietà da S-O diverse**
* **Multilevel: Persona (S-O) – Studente (SO-C) – Studente di Fisica (sottoclasse della sottoclasse)**

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

**Tutte le classi ereditano da System.Object. Se non viene specificata la classe da cui ereditare si eredita direttamente da Object. Alcuni metodi:**

* **ToString**
* **Equals**
* **Finalize**

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

**Il try/catch è il comando per la gestione delle eccezioni. Serve per gestire le eccezioni che potrebbero avvenire in fase di interazione con l’utente. Se immagino che in un blocco di codice possa verificarsi un’eccezione questo si inserisce nel blocco try e al verificrsi dell’eccezione si passa al blocco catch. Questo permette di non bloccare il flusso di esecuzione e di mostrare a schermo il modito dell’insorgere dell’eccezione, eventualmente specificando l’azione da fare nel caso si verifichi una specifica 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.