Esercitazione di Fine Settimana – Week 1

Nome Oriella

Cognome Vullo

1. Data 09/07/2021

Leggete attentamente ogni domanda e argomentare quanto più possibile fornendo anche degli esempi.

---

1. Quali sono i 3 tipi di runtime disponibili nella piattaforma .NET? Cosa è successo col rilascio di .NET 5?
   1) Framework 2) Core 3)Xamarin . Con .NET 5 si uniscono tutte le piattaforme che posso essere sviluppate negli stessi ambienti quali Visual Studio/Visual Studio for Mac/ Visual Studio Code.
   In questo modo si crea un solo standard con le stesse infrastrutture: linguaggi, compilaotre e runtime.

2. Quale è la funzione del Garbage Collector?
   Gestire della memoria che non viene utilizzata. Es. Quando un oggetto nel codice smette di esistere il GC libererà quello spazio prima associato ad esso.

3. Descrivere la differenza tra Value Type e Reference Type
   Il Reference Type è un puntatore verso l'aria di hype assegnata, tanto che posso assegnare a un Reference type un valore null, cioè che non punta a nulla ma occupa spazio.
   Al contrario il Value rappresenta direttamente lo spazio di memoria dato e per questo non può essere nullo.

4. Cos'è un Delegate?
   La definizione di un nuovo tipo dato che rappresenta una funzione.

5. Descrivere l'uso delle keyword `async / await`
   `Con async definiamo il meotodo che vogliamo sia Asyncrono, in questo modo il metodo, appena incontrerà la parola chiave await su una prozione di codice, lavorerà su un altro thread in modo da eseguirlo senza bloccare l'esecuzione del programma.`

6. Data una lista di istanze della classe `Votazioni`,

```
public class Votazioni
{
   public string Materia { get; set; }
   public string Studente { get; set; }
   public DateTime Data { get; set; }
   public int Voto { get; set; }
```

```
}
```

scrivere una query LINQ (Fluent oppure Query Expression) che restituisca un elenco di materie con il voto medio, quello più alto e quello più basso per ciascuna.

```
// Query Expression
var elencoMaterie =
                from v in voti
                group v by v.Materia into grp
                select new {
                    Nome = grp.Key,
                    Avg = grp.Average(dato => dato.Voto),
                    Max = grp.Max(dato => dato.Voto),
                    Min = grp.Min(dato => dato.Voto)
                };
```

7. Descrivere il Factory pattern e perché utilizzarlo
   Si crea un interfaccia dell'oggetto che verà esteso dagli oggetti concreti .
   Sarà poi la classe Factroy a crearlo a seconda delle informazioni ricevute.
   Lo utilizziamo per centralizzare la creazioen di un oggetto, fattorizzando il codice e rispettando così la divisione dei compiti.

Esercitazione Pratica

Realizzare una Console app (C#) che:

- Effettui il monitoraggio di una cartella in attesa di un file di testo con l'elenco delle spese (*spese.txt*)
- Apra e legga il file. Ogni riga è nel formato:

```
Data;Categoria;Descrizione;Importo
```

- Per ogni riga, determini se la spesa è approvata. Esistono diversi livelli di approvazione, a seconda dell'import della spesa
  - **Manager**: spese fino a 400€
  - **Operational Manager**: da 401€ fino a 1000€
  - **CEO**: sopra i 1000€
  - Nessuna spesa sopra i 2500€ è approvata

Usare il Chain of Responsibility pattern
(restituire il livello di approvazione)

- Per ogni spesa approvata, determini l'importo rimborsato sulla base della Categoria
  - **Viaggio**: 100% dell'importo + 50€ fisse
  - **Alloggio**: 100% dell'importo
  - **Vitto**: 70% dell'importo
  - **Altro**: 10% dell'importo

Usare il Factory pattern

- Salvi poi le informazioni sulle spese rimborsate e non rimborsate in un file di testo (*spese_elaborate.txt*)
  - Per ogni spesa rimborsata salvare una riga nel formato

```
Data;Categoria;Descrizione;APPROVATA;LvlApprov;ImportoRimborsato
```

  - Per ogni spesa non rimborsata salvare una riga nel formato

```
Data;Categoria;Descrizione;RESPINTA;-;-
```