## Prova Week 1

1. ***Descrivere le differenze tra Reference Type e Value Type***

I value type contengono direttamente il dato nell’ambito dello stack del thread. Una copia del value type implica la copia dei dati in esso contenuti e le modifiche hanno effetto solo sull’istanza corrente. Contengono sempre un valore e il valore null non è ammesso. I value type comprendono:

* Int, byte, bool, etc.
* Enum, struct 🡪 definiti dall’utente

I reference type contengono solo un riferimento ad un oggetto nell’ambito dell’heap. La copia di un reference type implica la duplicazione del solo reference e le modifiche su due reference modificano l’oggetto a cui puntano. Il reference che non referenzia alcuna istanza vale null, le classi son del tipo reference type.

1. ***Cosa è un delegate?***

Un delegate è l’equivalente .NET dei puntatori a funzione del c/c++ unmanaged, ma ha il vantaggio di essere tipizzato. Il compilatore crea una classe che deriva da System.Delegate oppure System.MulticastDelegate: sono due classi speciali che solo il compilatore può derivare. Da programma il delegate viene istanziato passandogli nel costruttore il nome del metodo di cui si vuole creare il delegate.

1. ***Definire come costruire il design pattern Factory*  e qual è la sua utilità**

E’ un modello che permette di risolvere il problema della creazione di oggetti senza specificare le classi concrete. Definisce un metodo che possiamo usare per creare un oggetto senza usare il suo costruttore: la cosa importante è che le sottoclassi possano sovrascrivere questo metodo e creare oggetti di diversi tipo. Si realizza nei seguenti passaggi:

1. Creazione dell’interfaccia o classe astratta che esporrà le operazioni principali
2. Creazione delle classi di oggetti
3. Creazione della classe Factory
4. ***Dare una panoramica degli elementi principali nel Framework .Net. Cosa è successo con l'introduzione di .Net 5?***

.NET Framework è l’ambiente di esecuzione runtime di .NET in cui vengono gestite le applicazioni destinate allo stesso. E’ costituito dal:

* Common Language Runtime 🡪 fornisce la gestione della memoria e altri servizi di sistema
* Libreria di classi 🡪 consente ai programmatori di sfruttare codice per tutte le aree principali dello sviluppo di applicazioni.

.NET 5 rispetto al .NET Framework e .NET Core supporta più tipi di app e più piattaforme.