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

Value type sono la maggior parte dei tipi primitivi (int, double, bool, char, DateTime) più enum e struct. Contengono direttamente il valore del dato e vengono allocati nello stack. Non vengono istanziati con new e non possono essere di valore null.

Reference type sono tutti i tipi di dati più complessi, come array (non so precisamente in c# ma in c una stringa è un vettore di char della quale abbiamo il puntatore all’elemento [0]), liste e in generale tutte le istanze di classi. Non contengono direttamente il valore dell’oggetto, ma un puntatore (o riferimento) ad esso, che viene istanziato nella heap. Ne consegue che copiandolo ottengo la copia del riferimento, e modificandolo ottengo la modifica all’oggetto puntato. Un reference type non istanziato ha valore null.

1. ***Cosa è un delegate?***

I delegate sono dei puntatori a metodi, dove viene definito il tipo di valore restituito e i parametri ammessi. Un esempio può essere:

delegate double Operazione (double primoOperando, double secondoOperando)

ed è compatibile con qualsiasi metodo che accetti 2 double in ingresso e restituisca un double, come:

double Somma (double primoAddendo, double secondoAddendo)

in questo modo posso istanziare un oggetto di tipo operazione:

Operazione y = new Operazione(Somma);

e richiamarlo passandogli i parametri che accetta Somma:

double risultato = y(x1,x2);

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

L’utilità di un design pattern è che ho già modelli di problemi ricorrenti. In particolare Factory è utile per definire una classe che funge da intermediario per creare oggetti di classi diverse. Una classe Cliente chiede a una classe Creatore di interfacciarsi con le classi Prodotto, creando il prodotto opportuno in base alle informazioni passate dal cliente.

1. ***Dare una panoramica degli elementi principali nel Framework .Net. Cosa è successo con l'introduzione di .Net 5?***

Il .net framework è l’ambiente che permette la creazione di applicazioni windows e web (asp.net) e contiene le librerie di classi e la macchina virtuale (clr) che traduce il linguaggio intermedio compilato (bytecode) in linguaggio macchina (codice nativo).

Con .net 5 vengono unificati gli ambienti di sviluppo windows con .net core (ambiente open source cross platform per lo sviluppo di universal windows app) e xamarin (sviluppo ios e android)