# Test di Valutazione-Week1

# 1.

# C# è un linguaggio tipizzato in quanto prevede l'utilizzo dei cosiddetti tipi come mezzo per descrivere le caratteristiche di una variabile o di un oggetto appartenente ad classe, e questo permette che questi ultimi vengano istanziati ed inizializzati in maniera coerente a come sono s stati dichiarati ed al loro obbiettivo.

# Possiamo due tipologie di Dati:

# Value Type, che comprende Tipo Primitivi(int, Boolean, String, Byte, ...), Enum e Struct(definiti dall'utente): per la variabile/ oggetto definisce il tipo e ne instanzia il valore, deve essere necessariamente inizializzata e non può essere nullo

# Reference Type: è un oggetto/variabile che rappresenta un riferimento ad un altro, ma non ne copia il valore; può essere nullo e deve essere instanziato.

# 2.

### Un Delegate è un Oggetto appartenete alla Classe Delegate, che permette di incapsulare in modo sicuro un metodo, esso è indipendente dla Type di ritorno di quest’ultimo: rappresentano una sorta di puntatore a funzione in C#: il delegate viene definito in questo modo

# *public* il *Type* del Delgate è *il tipo di ritorno* del *metodo incapsulato* che qui assume il ruolo di *parametro,* il Type è l’*argomento accettato da quest’ultimo(string).*

# I *metodi incapsulati* possono esser*e più di uno.*

# delegate

# void Del(string message);

# 3.

# Il Design Pattern Factory, è utile quando vogliamo istanziare una serie di Oggetti appartenenti ad una classe derivata da una Base, quindi che appartengono ad una stessa Categoria, ma possono avere caratteristiche specifiche diverse(valori diversi per una o più proprietà).

# Le caratteristiche comuni(attibuti e metodi) vengono definiti in un Interfaccia che le Classi Concrete devono seguire instanziano ed inizializzando le proprietà e sovrascrivendo i metodi

# Viene Creata una classe Factory che permette di ottente oggetti della Categoria definita in precendenza, che contiene:

# Tipo instanzia un oggeto di Typo Interfaccia ed assegna null;

# Classi Concrete che implemantano(override) i metodi contenuti nell’interaccia e definiscono gli attributi

# che contiente un metodo, con parametri(carateristiche specifiche), che permette di instanziare oggetti di Type Interfaccia, e con l’utilizzo di uno switch/if clause a seconda di esse(es: Animali, Mammiferi – Ovipari -OvoVivipari – Canguro – Rana -Vipera, parametri: faUova?(boolean), IsGestazioInterna(boolean)

# es: Animali – Insetti -Aracnidi -Funghi o Batteri, Gatto – Coccinella – Scorpione –

# Covid-19, parametri switch – numZampe, case 2 - 4 – 6 – 8 – (0 + isSnake? is Warm?) - Default)

# Es: Classe Astratta Animali

# Classi Derivate Mammiferi – Ovipari – OvoVivipari

# Classi Canguro

# Rana

# Vipera

# 