# Test di fine settimana – Week 3

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Eleonora |
|  |  | Cognome | Lombardo |
|  |  | Data | 04-06-2021 |

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. *Elencare le principali differenze tra strutture e classi.  
   La prima differenza è che l’assegnazione tra variabili struct ne fa una copia (della struttura) invece con le classi si fa una copia del puntatore a quell’oggetto ma l’oggetto rimane unico.  
   Le struct non possono avere un costruttore senza parametri e non può avere un distruttore.  
   Le classi invece possono avere costruttori con parametri e senza e possono avere distruttori. Le struct ereditano la classe System.ValueType invece le classi System.Object.*
2. *Nell’istruzione* ***Persona p = new Dipendente();*** *la variabile p si comporta in maniera polimorfa. Perché? Cosa si intende per comportamento polimorfico?  
   Si parla di polimorfismo quando gli oggetti di classe derivata possono essere trattati come oggetti della classe base e poi in fase di esecuzione scegliere in maniera dinamica il metodo adatto, questa cosa viene fatta attraverso i metodi virtual nella classe base e le classi derivate possono farne l’override. Verrà chiamato il metodo non virtual più vicino. Ovviamente i metodi delle classi base devono avere lo stesso nome di quello virtual della classe base.  
   Si comporta in maniera polimorfa perchè si è usato il costruttore della classe derivata Dipendente su un oggetto definito come Persona, questo insieme al meccanismo sopra descritto porta al polimorfismo.*
3. *In che modo è possibile utilizzare la combinazione di keyword del linguaggio virtual e override? Fornire un esempio.  
   La keyword override serve per implementare o estendere l’implementazione di un metodo astratto o virtuale con lo stesso nome. La keyword virtual invece fa si che il metodo non possa essere chiamato e “obbliga” le sottoclassi a doverlo implementare facendone l’ovveride per poterlo richiamare.  
   Un esempio in cui vengono usate insieme è nel caso del polimorfismo.   
   Ma anche quando in una classe si fa l’override del metodo ToString() di Object.*
4. *Spiegare la differenza tra Array, Collection e Collezioni Generiche.  
   Array sono un tipo specifico di Collection, un sottoinsieme. Gli array sono tipizzati e hanno dimensione fissa. Nel senso che inizializzandone uno, per esempio int [] array=new int[3], non sarà più possibile modificare la dimensione, per tutto il l’esecuzione la dimensione rimarrà 3.  
   Le collection invece non hanno dimesioni prefissate, possono variare dinamicamente ma rimangono fortemente tipizzati. Cioè per esempio la classe List<T> utilizza un template, ma una volta istanziata una lista List<int> l=new List<int>() rimarrà int per tutta l’esecuzione.  
   Le collection sono un sottoinsieme delle Collection Generics, quest’ultime non sono tipizzate e non sono di dimensione prefissata, gli elementi quindi possono essere di tipo eterogeneo e la dimensione crescere e diminuire dinamicamente.*

*Esercitazione pratica*

*Creare una Console App per la gestione di un carrello di un e-commerce.*

*Il sito ha degli utenti iscritti. (Inserire almeno un utente già iscritto).*

***L’utente*** *è definito con*

* ***Username,***
* ***Password,***
* ***Nome e Cognome****.*

*L’e-commerce prevede che vengano messi in vendita prodotti d’abbigliamento, alimentari e di elettronica tutti caratterizzati da:*

* *un* ***codice****,*
* *una* ***descrizione****,*
* *il* ***prezzo****,*
* *una* ***percentuale di sconto****.*

*I prodotti di tipo abbigliamento specificano anche una* ***taglia*** *(string) e il* ***brand*** *(string), i prodotti alimentari hanno anche una* ***data di scadenza*** *mentre i prodotti di elettronica un* ***produttore*** *(string.*

*Per ogni utente è previsto un solo carrello contenente le righe di dettaglio dell’ordine e il prezzo totale da pagare (nota: è importante risalire dall’utente al carrello e non il viceversa).*

*Ciascuna riga di dettaglio contiene le seguenti caratteristiche:*

* ***prodotto****,*
* *la* ***quantità ordinata****,*
* *il* ***prezzo totale*** *al netto dello sconto (rispetto alla quantità e al prezzo “pieno” del prodotto),*
* *il* ***prezzo totale scontato*** *(calcolato rispetto alla percentuale di sconto del singolo prodotto).*

*All’accesso, viene chiesto username e password. Se sono corrette si accede al menu.*

1. *Aggiungi prodotto al carrello\**
2. *Elimina prodotto*
3. *Modifica la quantità di un prodotto già inserito*
4. *Stampa a video riepilogo del carrello dell’utente (formato a piacere)*
5. *Esci*

*\*Nel caso sia inserito un prodotto che già esiste nel carrello questo va a modificare la quantità del prodotto precedentemente inserito.*

*Requisiti tecnici:*

*- Rappresentare opportunamente la gerarchia dei prodotti.*

*- Utilizzare la collection adeguata considerando che ogni prodotto ha un codice univoco.*

*Consigli:*

*Creare metodi ad hoc per creare le entità di partenza “precaricate” nell’e-commerce.*

*Mettere la prova pratica e teorica su Github.*