# Test di fine settimana – Week 3

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome |  |
|  |  | Cognome |  |
|  |  | Data |  |

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.*

***Le classi supportano i costruttori, supportano l’ereditarietà, supportano metodo finalize e sono refernce type.***

***Le strutture non hanno costruttori, né di default e né con l’inizializzazione; non supportano ereditarietà; non hanno metodo finalize; è un value type.***

1. *Nell’istruzione* ***Persona p = new Dipendente();*** *la variabile p si comporta in maniera polimorfa. Perché? Cosa si intende per comportamento polimorfico?*

***Si comporta in maniera polimorfa perché Dipendente è sottoclasse di Persona. Il comportamento polimorfo consiste nel fatto che l’istanza di un tipo si comporti come se fosse istanza di un altro tipo. Il polimorfismo è subordinato all’esistenza di una relazione di derivazione tra i due tipi***

1. *In che modo è possibile utilizzare la combinazione di keyword del linguaggio virtual e override? Fornire un esempio.*

***La keyword “virtual” è usata nella classe base per modificare metodi e proprietà. Con “override” si può estendere o modificare un metodo virtual o abstract;***

***public virtual void descrizione();***

***[...]******public override void descrizione() {***

***String s = 11;}***

1. *Spiegare la differenza tra Array, Collection e Collezioni Generiche.*

***Un array ha una grandezza pdefinita dall’utente, al contrario di una collezione che si può espandere aggiungendo elementi. Inoltre, le collezioni hanno metodi che permettono la rimozione, aggiunta, ricerca ecc... di un elemento.***

***Nelle Collections noi possiamo inserire tipi generici mentre tramite System.Collections.Generics dobbiamo inserire solo tipi di collezioni T (Collection<T>):***

*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.*