# Esercitazione di Fine Settimana – Week 2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Tiziana |
|  |  | Cognome | Camboni |
|  |  | Data | 02/04/2021 |

Leggete attentamente ogni domanda e argomentare quanto più possibile fornendo anche degli esempi.

1. Descrivere le modalità per ritornare più valori da un metodo in C#

I metodi in c# possono ritornare i valori principalemnte in tre modi:

* Void: non viene ritornato alcun valore
* MyType: viene dichiatato il tipo di cui si vuole il ritorno prima del Nome del metodo;
* Tuple: le tuple vengono utilizzate quando si vuole in ritorno multiplo, fino ad 8 valori. I tipi di ritorno vengono dichiari tra parentesi prima del nome del metodo.

1. Descrivere le due tipologie di casting tra tipi in C#

Il casting può avenire in modo implicito quando il tipo finale della conversione è meno stringente del tipo di partenza e viene fatto con un’assegnazione

(es. int x= 1000;

dec y= x;

il casting puo avenire in modo espliciti con l’utilizzo delle parentesi che racchiudono il tipo di arrivo, posizionate prima della variabile da convertire.

1. Quali sono gli utilizzi della keywork static?

Gli utilizzi della keywork static possono fare riferimento a classe, metodo, campo;

* la classe statica non viene utilizzata come un oggetto (che viene istanziato) ma per esempio, può essere un appoggio per metodi.
* il metodo statico ha un funzionamento a se stante senza avere bisongno di un’istanza della classe
* proprietà statica

1. Descrivere le differenze tra le interfacce utilizzabili per le collection

Le interfacce utilizzabili per le collection permettono di

1. Cos’è un Generic?

Un generic è definibile come un “tipo generico”in c#, dichiarato come “ T”, con il vantaggio prinicipale che permette di scrivere del codice utilizzabile con diversi tipi, quindi un riuso di codice. Un metodo in che prende in input un parametro T infatti può essere utilizzato ad esempio prendendo in input un intero.

**Esercitazione Pratica**

* Realizzare una classe Account per gestire un conto bancario, con le seguenti proprietà:
  + *Numero di Conto*
  + *Nome della Banca*
  + *Saldo*
  + *Data Ultima Operazione*
  + *Lista di Movimenti*
* Realizzare l’overload degli operatori + e – in modo che sia possibile aggiungere movimenti attivi e passivi alla lista dei movimenti (l’overload dovrà anche occuparsi di aggiornare il Saldo e la Data di Ultima Operazione)

Note: sovrascrive balance in base al movimento,

account + movimento -> rende account

* Realizzare un metodo Statement() che stampi i dati del conto, inclusa la lista dei movimenti
* Realizzare una gerarchia di classi per rappresentare movimenti bancari (Movement). Tutte le classi avranno le proprietà
  + *Importo*
  + *Data del Movimento*
  + Realizzare le classi che rappresentano:
    - CashMovement, con la proprietà aggiuntiva *Esecutore*
    - TransfertMovement, con le proprietà aggiuntive *Banca d’Origine* e *Banca Destinazione*
    - CreditCardMovement, con le proprietà aggiuntive *Tipo* (enum con i valori AMEX, VISA, MASTERCARD, OTHER) e *Numero di Carta*
  + Tutte le classi saranno dotate di costruttore che accetti tutti i parametri necessari per popolare le proprietà
  + Tutte le classi dovranno implementare la propria versione del metodo ToString() e visualizzare tutti i dati
* Realizzare una Console app che
  + Crei un nuovo Account
  + Permetta di inserire diversi tipi di Movimenti (input dall'utente)
  + Stampi i dati del conto e i movimenti