# Esercitazione di Fine Settimana – Week 2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Nome | Martina |
|  |  | Cognome | Libreri |
|  |  | Data | 2/4/2021 |

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#

Pre tornare più valori da un metodo vengono utilizzate diverse modalità. É possibile utilizzare la parola chiave out al’interno della parentesi con i parametri di ingresso del metodo non void (ossia con un parametro di ritorno), per esempio in una funzione che al proprio interno deve svolgere un’operazione e oltre al parametro di ritorno dell’operazione restituisce un booleano per verificare che l’operazione sia stata svolta.

Public static bool Operazione(int a, out int b)

Mentre un secondo metodo per la restituzione di più parametri è l’utilizzo della tupla: che permette di restituire fino a otto parametri.

Altrimenti un altr metodo è quello di restituire una classe (o struttura) con i vari parametri utilizzati da essa.

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

Il casting può essere svolto in due modalità : esplicito ed implicito.

Il casting implito viene svolto su tipo che possono svoltere l’operazione di casting senza perdere informazione come ad esempio un intero che viene convertito in un long.

Il casting implicito richiede una determinata scrittura e nell’operazione potranno essere persi alcuni dati oppure potrà anche non avvenire.

Un esempio: convertire un float in un int 🡪 int a; float b = ....; a = (int)b;

1. Quali sono gli utilizzi della keywork static?

La keyword static viene utilizzata per classi, metodi e campi. Viene utilizzata quando non è richiesto che venga istituita un’istanza dell’oggetto.

Un metodo statico non appartiene alla classe quindi non è parte di nessun oggetto, non ha accesso alle variabili di istanza di nessun oggetto e può essere invocato anche se non ho nessuna instanza di una classe.

I metodi non statici sono metodi propri della classe, se esistono due metodi che svolgono la stessa operazione ma un metodo è statico e l’altro no, questi possono essere chiamati con lo stesso nome in quanto avranno parametri di input differenti.

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

Per le collection è possibile utilizzare differenti tipi di interfacce:

* IEnumerable : definisce un metodo GetEnumerator che ritorna il valore successivo della collection.
* IList : è possibile accedere agli elementi tramide index per poterlo rimuovere o inserire in una determinata posizione
* IDictionary : i valori all’interno della collection sono definiti da chiave-valore e è possibile accedere ad un elemento tramite chiave.
* ICollection : utilizza i metodi Add,Remove per inserire e rimuovere elementi dalla collection e tramite il count è possibile avere il numero di elementi che lo compongono.

1. Cos’è un Generic?

I generics sono costrutti del linguaggio che permettono di scrivere classi e metodi indipendentemente dal tipo, avendo la possibilità quindi di utilizzare quella classe/metodo per tipi differenti. In questo modo ho la possibilità di scrivere meno codice e riutilizzarlo.

Ha la caratteristica di utilizzare dopo il mone della classe “<T>”,ma non è possibile assegnare al generics un valore null perchè potrebbe essere utilizzato anche per dei value type in quanto non è un reference type. Possono anche essere utilizzati dei constraint per utilizzare quel generico attraverso l’inserimento del Where nella definizione della classe.

Public class ClassName<T> where T : class 🡪 può essere utilizzato solo per i tipi classe.

**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)
* 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