Alessia Gualtieri 23/1/2021

**TEST WEEK 2**

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

Esistono tre diversi modi per ritornare più valori in C#:

**REF**: Passando per riferimento una variabile tra i parametri, questa può essere modificata all’interno del metodo e una volta terminata l’esecuzione di tale metodo tale valore resta modificato anche nel chiamante del metodo. Si deve specificare la keyword **ref** al parametro che vuole essere restituito.

Esempio:

public static void Menu()

{

//...

Int choice = 0; //qui choice vale 0

bool letto = LeggiSceltaUtente(**ref** choice);

Console.WriteLine(choice); //qui choice vale 3

}

public static bool LeggiSceltaUtente(**ref** int choice)

{

Console.WriteLine(choice); //choice = 0;

choice = 3; //qui choice vale 3

return true;

}

**OUT**: Analogo al ref, ma non è necessaria l’inizializzazione della variabile. Si deve specificare la keyword **out** nel parametro passato che vuole essere restituito.

Esempio:

public static void Menu()

{

//...

Int choice;

bool letto = LeggiSceltaUtente(**out** choice);

Console.WriteLine(choice); //qui choice vale 3

}

public static bool LeggiSceltaUtente(**out** int choice)

{

choice = 3; //qui choice vale 3

return true;

}

**TUPLA:** serve per restituire più valori direttamente tra i parametri restituiti, cioè non avviene una modifica dei valori passati come input ma sono segnati più valori come output nel metodo. Rende più chiara la divisione tra i parametri di input e quelli di output, se i valori non vengono usati per ciò che valgono all’interno del metodo è conveniente restituirli come variabili nuove.

Esempio:

public static void Menu()

{

(int min, int sms, int gb) = RestituisciValori(sim);

}

public static (int min, int sms, int gb) RestituisciValori(Sim sim)

{

return (sim.Min, sim.Sms, sim.Gb);

}

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

Si può fare un cast implicito e un cast esplicito tra tipi.

Il **cast implicito** si può fare quando non c’è perdita di informazione tra un tipo e l’altro, e avviene quando il tipo di origine è composto da un sottoinsieme di valori assumibili anche dal tipo di arrivo. Un esempio è il cast di un int in double. Non è necessario inserire alcuna keyword, basta fare l’assegnazione.

Esempio:

int x = 3;

double y = x;

Il **cast esplicito** diversamente avviene quando la conversione può portare alla perdita di informazione, per esempio quando si passa da un insieme più grande ad un suo sottoinsieme. In tal caso è necessario forzare il cast inserendo tra parentesi tonde il tipo a cui si vuole convertire tale valore.

Esempio:

double x = 3.4;

int y = (double)x; //Ho perdita di informazione, qui y = 3

1. **Cos’è il Garbage Collector e come funziona la sua attività**

Il Garbage Collector è uno strumento che serve per liberare la memoria da informazioni non più utilizzate. Cancella dalla memoria dati che non sono più in uso. Il suo intervento non avviene in un momento della compilazione specifico, e ha un funzionamento non regolarizzabile dal programmatore. Usa il metodo Finalize presente nella classe Object per capire come procedere alla cancellazione dei dati, è possibile fare l’override di tale metodo per classi che lo necessitano, altrimenti viene richiamato in automatico quello di Object.