



Empowering Digital Skills For The Jobs Of The Future



by



Academy .NET

Garbage Collector

Sommario

- Garbage Collection
- Rilascio esplicito delle risorse
- IDisposable
- Il metodo Dispose
- Il metodo Finalize

Garbage Collector

Gli oggetti istanziati in C#, in quanto allocati, utilizzano una parte della **memoria**.

Il **Common Language Runtime** dispone del servizio **Garbage Collector** che permette una gestione automatica della memoria occupata dagli oggetti.

Garbage Collector

Il **Garbage Collector** è basato su un algoritmo che verifica gli oggetti non referenziati.

Gli oggetti non referenziati sono a questo punto eliminati per liberare memoria.

Il servizio **Garbage Collector** viene avviato periodicamente dal **CLR**.

Garbage Collector

Se gli oggetti includono risorse **non gestite** è necessario il rilascio esplicito delle risorse quando non utilizzate.

Alcuni tipi di risorse non gestite sono oggetti che eseguono il wrapping delle risorse del OS come **file, connessioni di rete, connessioni di database**.

Rilascio esplicito delle risorse

Implementazione schema **Dispose**:

- Implementare interfaccia **IDisposable**
- Chiamata a **Dispose** quando le risorse non sono necessarie
- Dispose rilascia le risorse non gestite

IDisposable

L'interfaccia **IDisposable** richiede l'implementazione di un singolo metodo **Dispose**

```
protected virtual void Dispose(bool disposing)
{

}
}
```

Dispose - Implementazione

Il metodo Dispose eseguirà la pulizia degli oggetti:

```
class MyClass : IDisposable
{
    bool disposed = false;

    public void Dispose()
    {
        // Dispose of unmanaged resources.
        Dispose(true);
        // Suppress finalization.
        GC.SuppressFinalize(this);
    }
}
```

continua nella slide successiva...

Dispose - Implementazione

```
...  
protected virtual void Dispose(bool disposing)  
{  
    if(disposed) return;  
  
    if(disposing)  
    {  
        Console.WriteLine("Freeing managed resources");  
    }  
  
    Console.WriteLine("Freeing unmanaged resources");  
  
    disposed = true;  
}  
}
```

Finalize

Il metodo **finalize()** è definito nella classe **object**.

A differenza di **Dispose**, che viene chiamato dall'utente, **Finalize()** viene chiamato dal **garbage collector** quando il riferimento di un oggetto non viene utilizzato da molto tempo.

Garbage Collector

Il metodo **finalize** deve essere implementato manualmente.

Il **garbage collector** invoca il metodo **finalize()** poco prima di distruggere completamente l'oggetto.

```
protected void finalize()  
{  
    // finalization code here  
}
```

Differenze Dispose – Finalize

- Il metodo **dispose()** è definito nell'interfaccia **IDisposable**.
- Il metodo **finalize()** è definito nella classe **object**.
- Il metodo **dispose()** deve essere invocato manualmente all'interno del codice, mentre il metodo **finalize()** viene automaticamente invocato dal **garbage collector** prima che distrugga l'oggetto.

Differenze Dispose – Finalize

- Il metodo **dispose()** è più responsivo e libera l'oggetto istantaneamente, quindi non influisce sulle prestazioni.
- Il metodo **finalize()** è più lento e non libera istantaneamente le risorse detenute dall'oggetto.

Domande & approfondimenti

Academy .NET