## Erklärung

[Entwurfsmuster](https://de.wikipedia.org/wiki/Entwurfsmuster) der Kategorie der [Verhaltensmuster](https://de.wikipedia.org/wiki/Verhaltensmuster_(Software)).  
  
**Entwurfsmuster**: bewährte Lösungsschablonen für wiederkehrende Entwurfsprobleme.  
[**Verhaltensmuster**](https://de.wikipedia.org/wiki/Verhaltensmuster_(Software))**:** modellieren komplexes Verhalten der Software. 🡪 Erhöht die Flexibilität der Software hinsichtlich ihres Verhaltens.

Anstatt einen [Null-Verweis](https://en.wikipedia.org/wiki/Null_reference) zu verwenden, um die Abwesenheit eines Objekts (zum Beispiel eines nicht existierenden Kunden) zu vermitteln, verwendet man ein Objekt, das die erwartete Schnittstelle implementiert, dessen Methodenkörper jedoch leer ist. Der Vorteil dieses Ansatzes gegenüber einer funktionierenden Standardimplementierung besteht darin, dass ein Null-Objekt sehr vorhersehbar ist und keine Nebenwirkungen hat: **Es tut *nichts*.**

Zum Beispiel kann eine Funktion eine Liste von Dateien in einem Ordner abrufen und auf jeder eine Aktion ausführen. Im Falle eines leeren Ordners kann eine Antwort eine Ausnahme auslösen oder eine Nullreferenz anstelle einer Liste zurückgeben. Daher muss der Code, der eine Liste erwartet, verifizieren, dass er tatsächlich eine solche hat, bevor er fortfährt, was den Entwurf komplizieren kann.

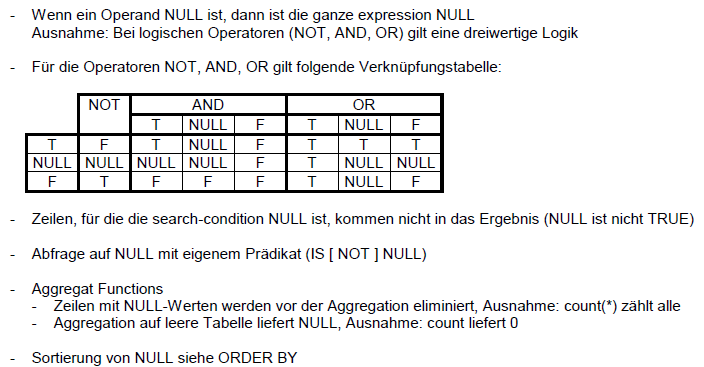
Wenn Sie stattdessen ein NULL-Objekt (d.h. eine leere Liste) zurückgeben, müssen Sie nicht überprüfen, ob der Rückgabewert tatsächlich eine Liste ist. Die aufrufende Funktion kann die Liste einfach als normal iterieren und tut nichts. Es ist jedoch immer noch möglich zu prüfen, ob der Rückgabewert ein Nullobjekt (eine leere Liste) ist und auf Wunsch anders reagiert.

Das Null-Objektmuster kann auch als Kontrollabschnitt für das Testen verwendet werden, wenn ein bestimmtes Feature wie eine Datenbank nicht zum Testen verfügbar ist.  
Andere Variante: [*Singleton*](https://de.wikipedia.org/wiki/Singleton_(Entwurfsmuster)) (Einzelstück) 🡪 weniger Speicher weil nur noch eine Instanz des Nullobjekts instanziiert wird

## +/-

* Vorteil:
* Es wird eine Code-Vervielfältigung vermieden.
* Es werden Fehler verhindert, wenn keine Überprüfung stattfand.
* Nachteil:
  + Ein Nullobjekt, das an die Stelle des Werts null treten soll, muss die öffentlichen Methoden der entsprechenden Schnittstelle implementieren.

## NULL und dreiwertige Logik



## Kritik

Dieses Muster sollte sorgfältig verwendet werden, da es Fehler / Fehler als normale Programmausführung erscheinen lassen kann.

Es sollte darauf geachtet werden, dieses Muster nicht zu implementieren, nur um Null-Überprüfungen zu vermeiden. Das übliche Muster in den meisten Sprachen mit Referenztypen besteht darin, eine Referenz mit einem einzelnen Wert zu vergleichen, der als Null bezeichnet wird. Außerdem muss zusätzlich getestet werden, dass kein Code irgendwo jemals null anstelle des Null-Objekts zuweist, da dies in den meisten Fällen und Sprachen mit statischer Typisierung kein Compiler-Fehler ist, wenn das Null-Objekt einen Referenztyp hat.