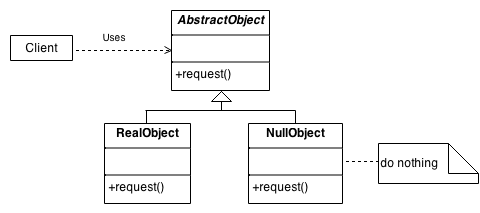
A behavioral design pattern with no behaviors.

The intent of a Null Object is to encapsulate the absence of an object by providing a substitutable alternative that offers suitable default do nothing behavior. In short, a design where "nothing will come of nothing"

Use the Null Object pattern when

* An object requires a collaborator. The Null Object pattern does not introduce this collaboration--it makes use of a collaboration that already exists
* some collaborator instances should do nothing
* you want to abstract the handling of null away from the client

**Class Diagram**



When component A used component B, it typically assumes that B is non null.

* You inject B, not some Option<B> type
* You do not check for null on every call

There is no option of telling A not to use an instance of B

* Its use is hard coded

Thus, we build a no-op, non-functioning inheritor of B(or some interface that B implements) and pass it to A

So Null Object is a no-op object that conforms to the required interface satisfying a dependency requirement of some object