Null:

**Other name (if any)**

placeholder

**What it does**

In Null Object pattern, a null object replaces check of NULL object instance. Instead of putting if check for a null value. Creating the same class to handle null values. Replace null return type with this class of object in the actual class. In the actual class don't check if the object is nullable or not (this will be replaced by the null object)

**Where to use**

placeholder

**Steps**

1. Abstract Object Class: Create an abstract object class with necessary functions.  
2. Concrete Object Class: It will extend the abstract function of the abstract object class.  
3. Concrete Null Object Class: It will extend the abstract function of the abstract object class with default value.  
4. Factory Class: There will be a function. If certain condition meets it will return the concrete object class otherwise will return null object class.  
5. Client Code: Create object using abstract object class and factory class. Pass a type to the specific function of the factory class. Call the specific function of the concrete object class/concrete null object class.

**Special cases (if any)**

placeholder

**Advantages**

1. Clients can treat real collaborators and null collaborators uniformly. Clients normally don’t know whether they’re dealing with a real or a null collaborator. This simplifies client code, because it avoids having to write testing code which handles the null collaborator specially.

**Disadvantages**

1. Can necessitate creating a new NullObject class for every new AbstractObject class.

**Code**

Tutorialspoint

**Difference with similar pattern**

placeholder

**Diagram**

Tutorialspoint