

NULL OBJECT

Design Pattern

"nothing will come of nothing"

Mihai Suci
30432

INTRODUCTION

- It is a behavioral pattern
- 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.



USAGE

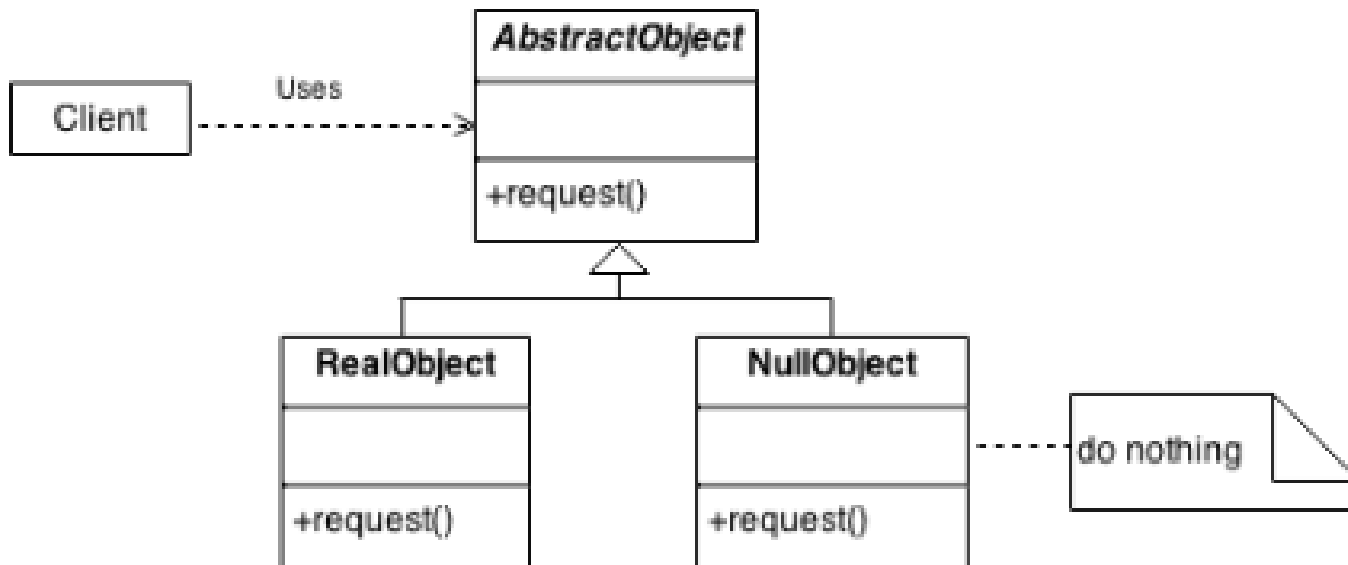
When

- an object requires a collaborator
- some collaborator instances should do nothing
- you want to abstract the handling of null away from the client



STRUCTURE

- Client - requires a collaborator
- AbstractObject - declares the interface for Client's collaborator
 - implements default behavior for the common interface
- RealObject - defines a concrete subclass of AbstractObject whose instances
 - provide useful behavior that Client expects
- NullObject - implements its interface to do nothing
 - provides an interface identical to AbstractObject's so that a null object can be substituted for a real object



EXAMPLE

```
class Application {  
    private PrintStream debugOut;  
    public Application(PrintStream debugOut) {  
        this.debugOut = debugOut;  
    }  
  
    public void doSomething() {  
        int sum = 0;  
        for (int i = 0; i < 10; i++) {  
            sum += i;  
            debugOut.println("i = " + i);  
        }  
        System.out.println("sum = " + sum);  
    }  
}
```

```
class NullOutputStream extends OutputStream {  
    public void write(int b) {  
        // Do nothing  
    }  
}  
  
class NullPrintStream extends PrintStream {  
    public NullPrintStream() {  
        super(new NullOutputStream());  
    }  
}
```

```
public class NullObjectDemo {  
    public static void main(String[] args) {  
        Application app = new Application(new NullPrintStream());  
        app.doSomething();  
    }  
}
```

sum = 45

ADVANTAGES & DISADVANTAGES

○ Pros

- It defines class hierarchies consisting of real objects and null objects.
- Also makes the client code simple. Clients can treat real collaborators and null collaborators uniformly.

○ Cons

- Can be difficult to implement if various clients do not agree on how the null object should do nothing as when your AbstractObject interface is not well defined.
- Can necessitate creating a new NullObject class for every new AbstractObject class.

