

# Using Test Doubles



**Emily Bache**

Technical Coach

@emilybache | [coding-is-like-cooking.info](http://coding-is-like-cooking.info)

# Overview/ Summary

## Kinds of Test Double:

- Stubs
- Fakes
- Dummies
- Spies
- Mocks

**Use a test double when:**

**The unit you want to  
test has a dependency**

**And for some reason you  
don't want to use the real one**

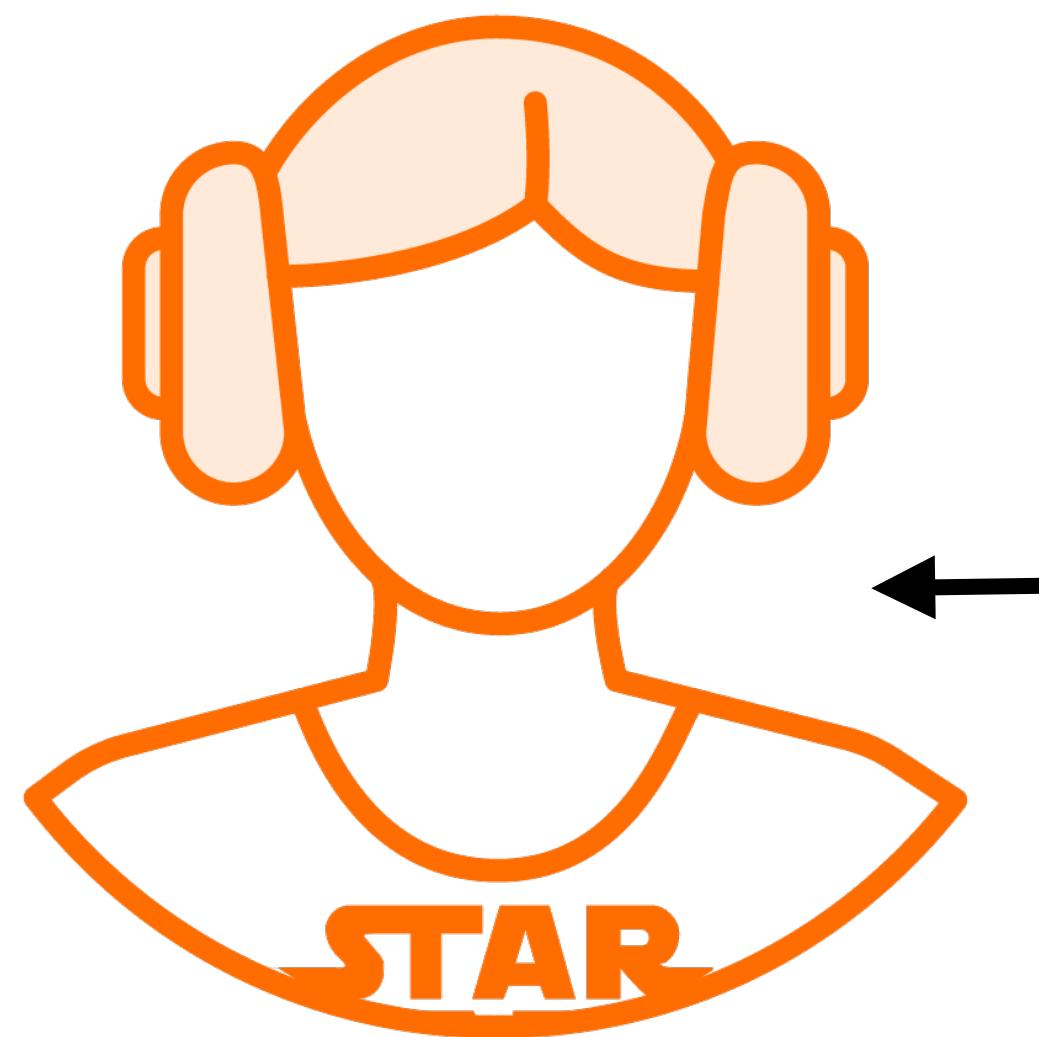
**The unit you want to test**

**Other Code it uses**

**Replace it with a Test Double**

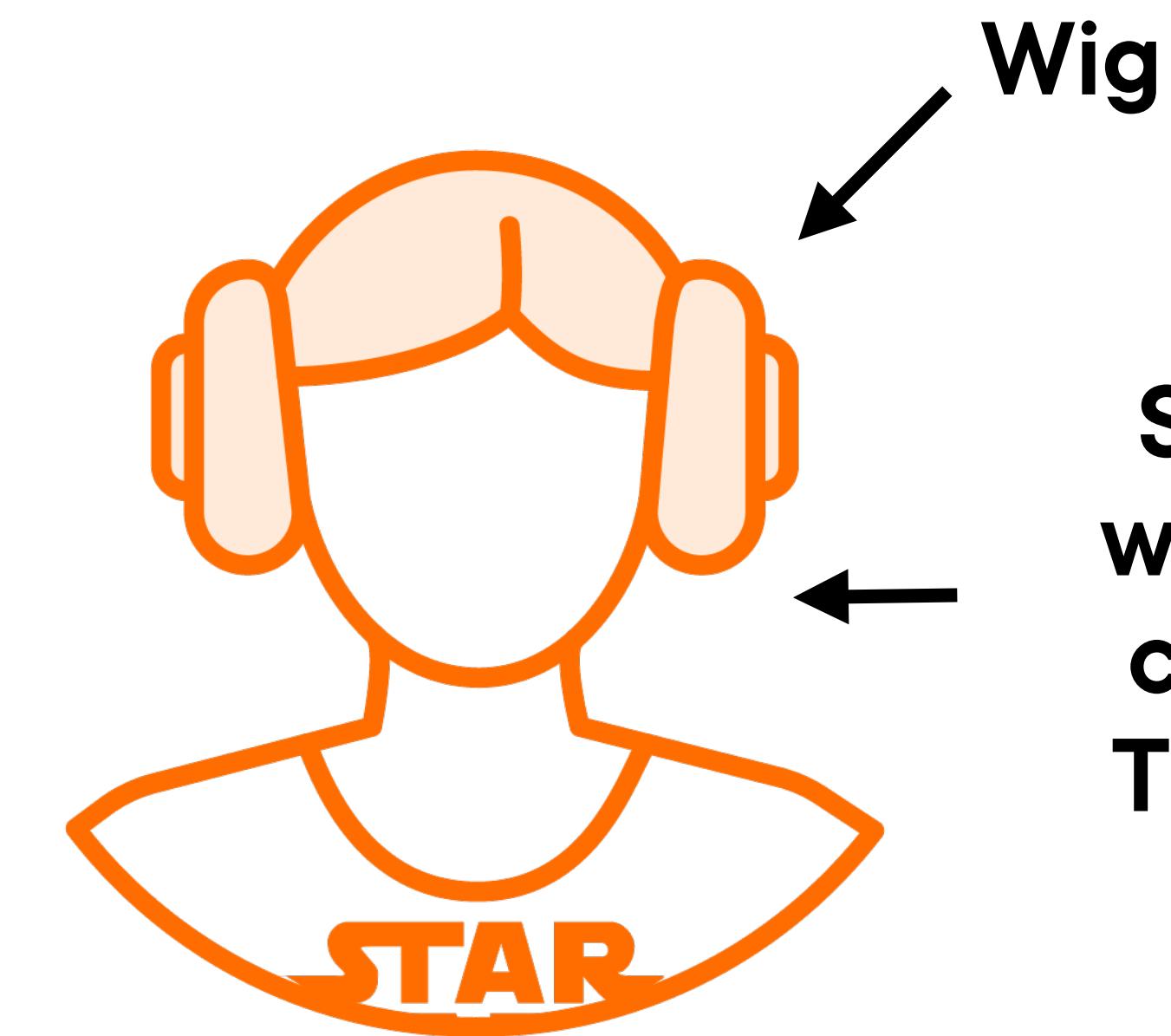
# Test Double - Like a Stunt Double

The real actor



Famous  
actress

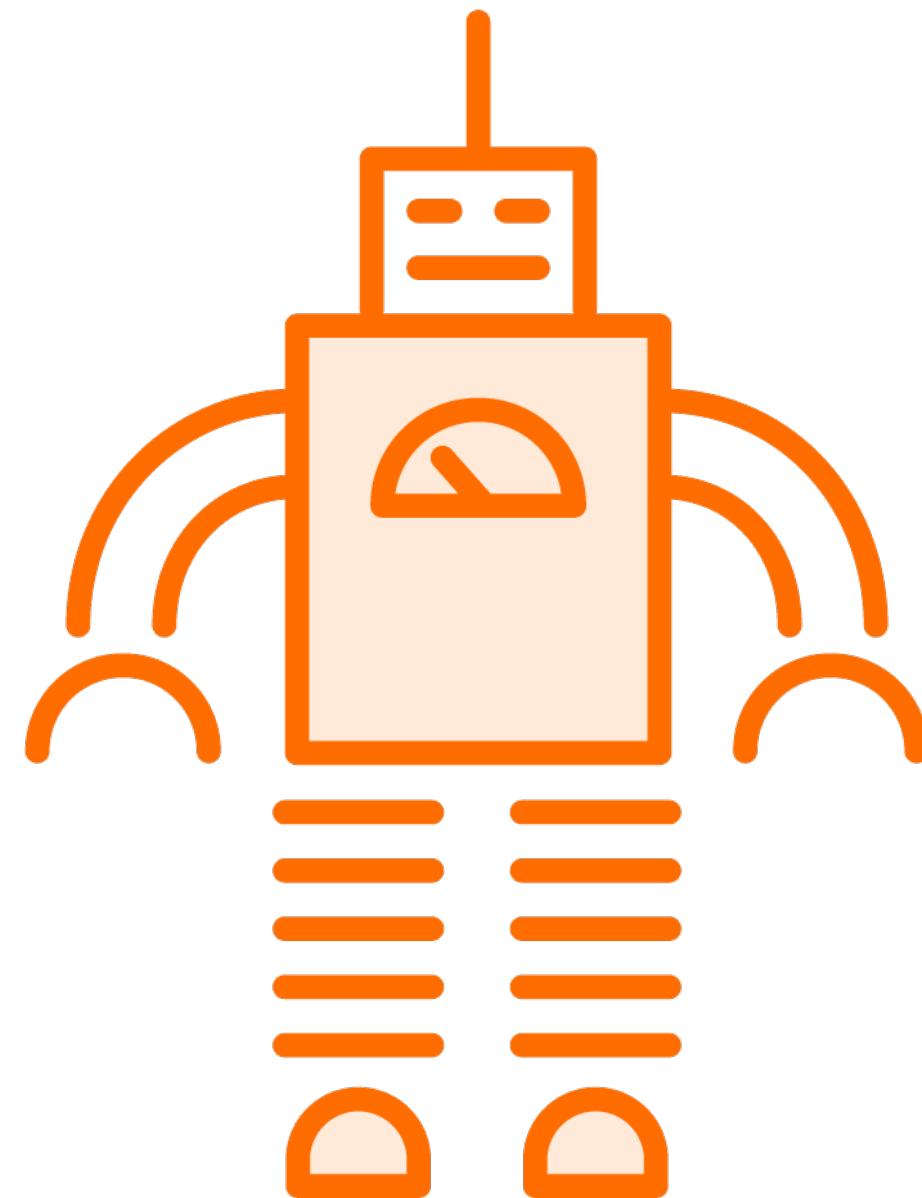
The stunt double



Some  
woman  
called  
Tracey

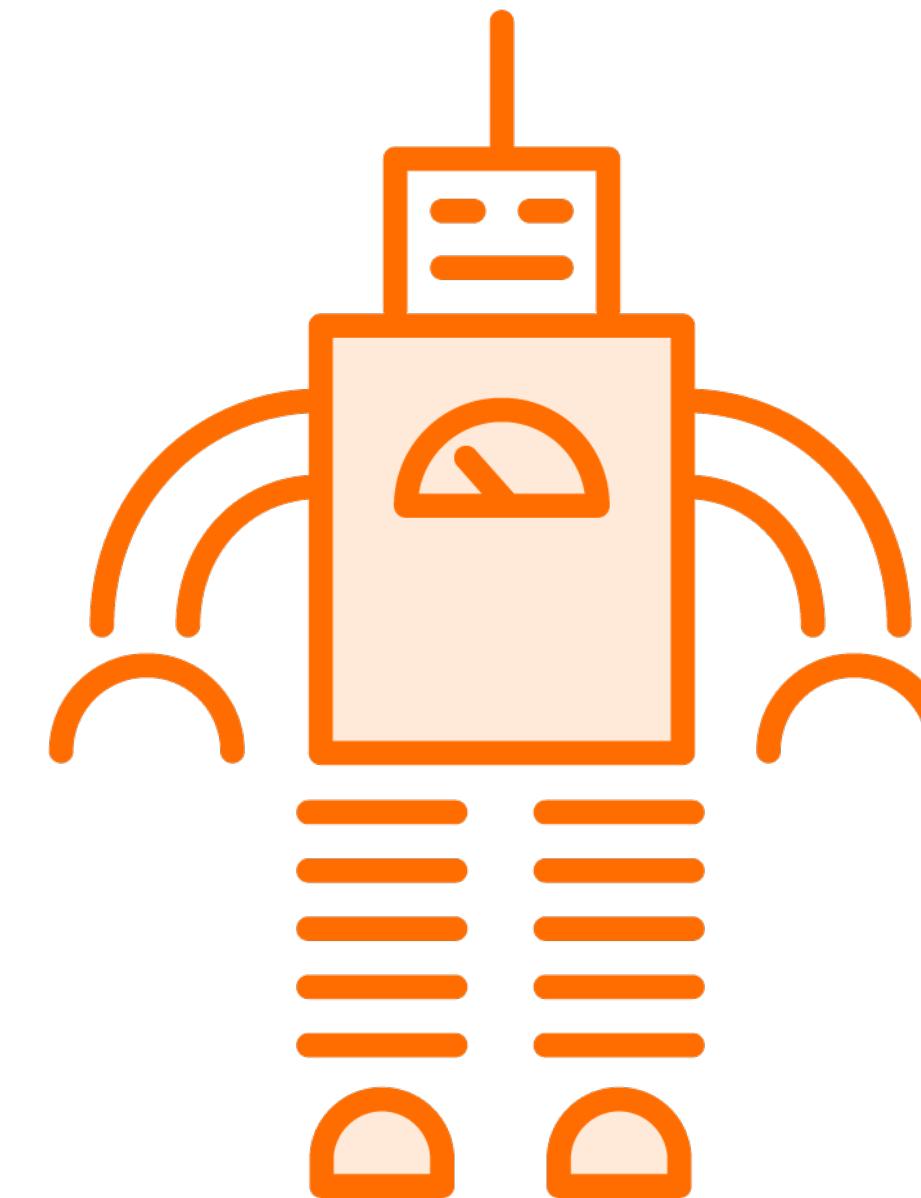
# Test Double - Like a Stunt Double

The real object



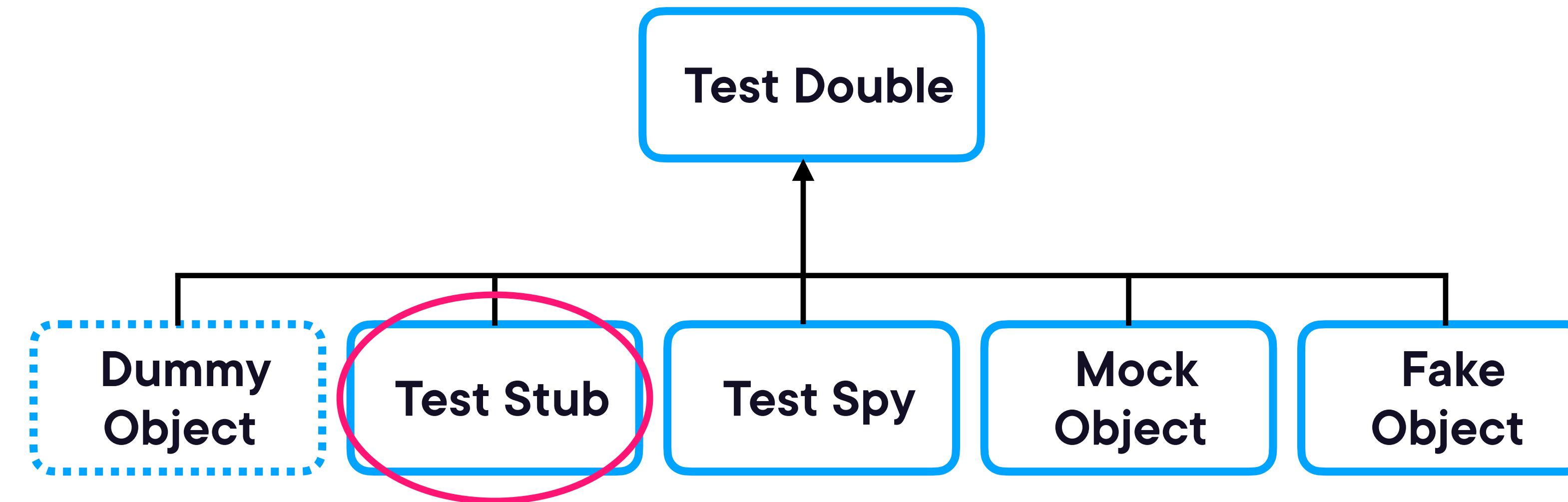
Complex,  
production  
logic

The test object



Everything  
it does is  
controlled  
by the test

# Different Kinds of Test Double - Stub

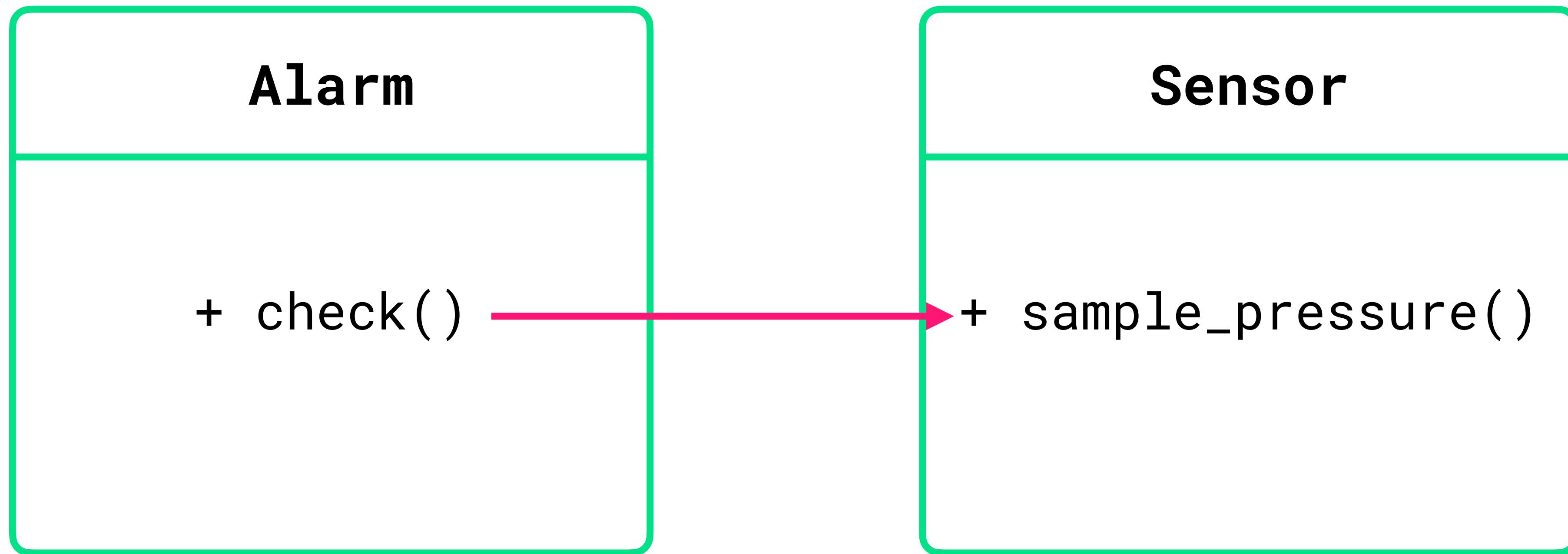
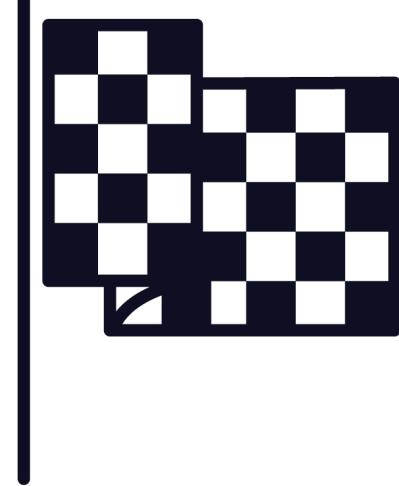
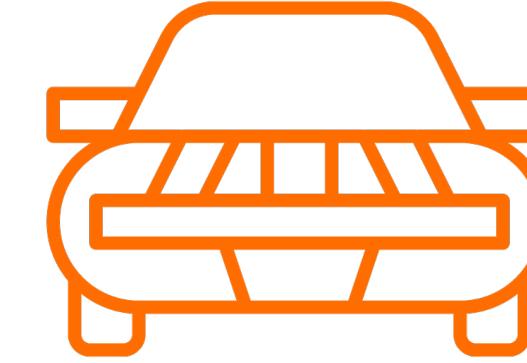


From <http://xunitpatterns.com>

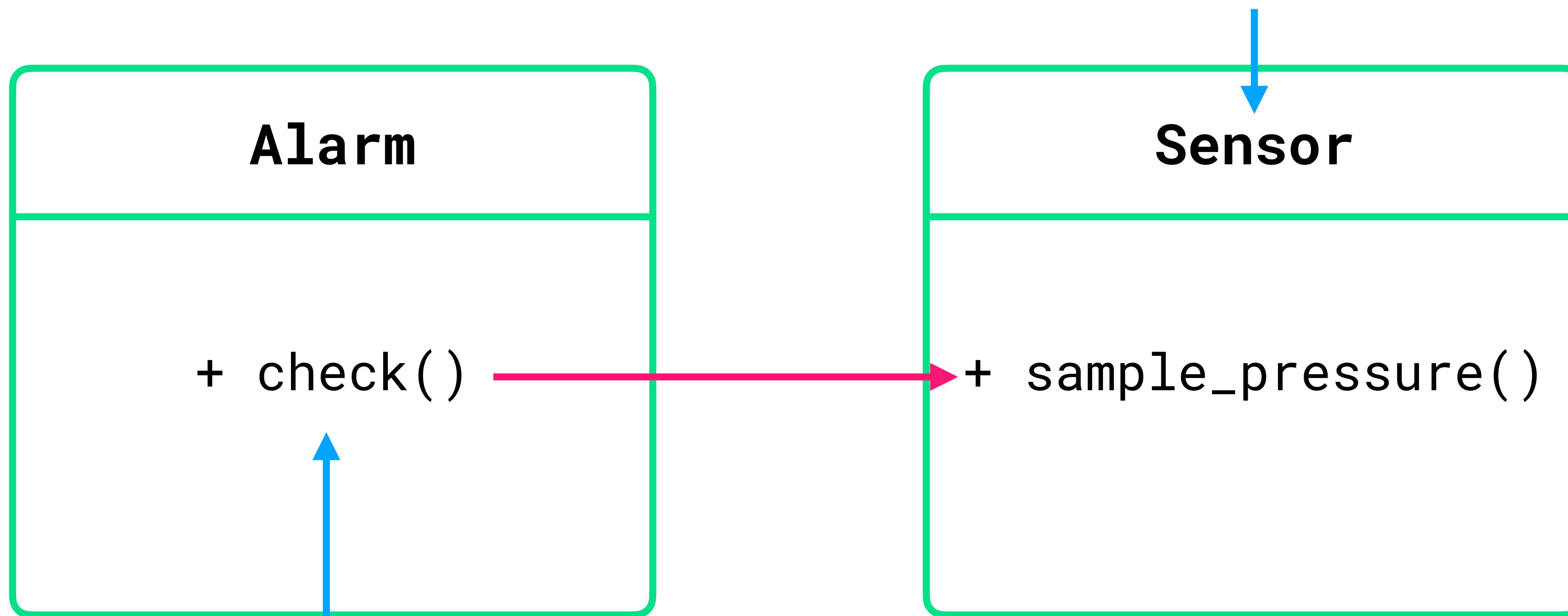
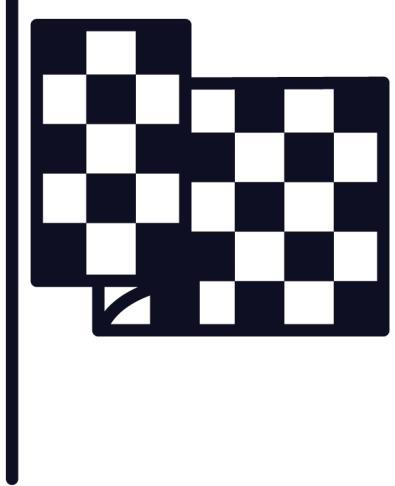
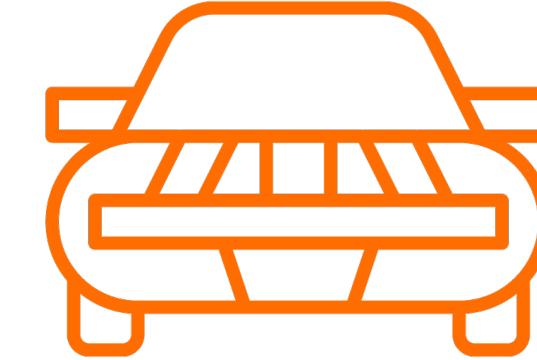
See also: [https://en.wikipedia.org/wiki/Test\\_Double](https://en.wikipedia.org/wiki/Test_Double)

## **Using a Stub**

# Racing Car Example



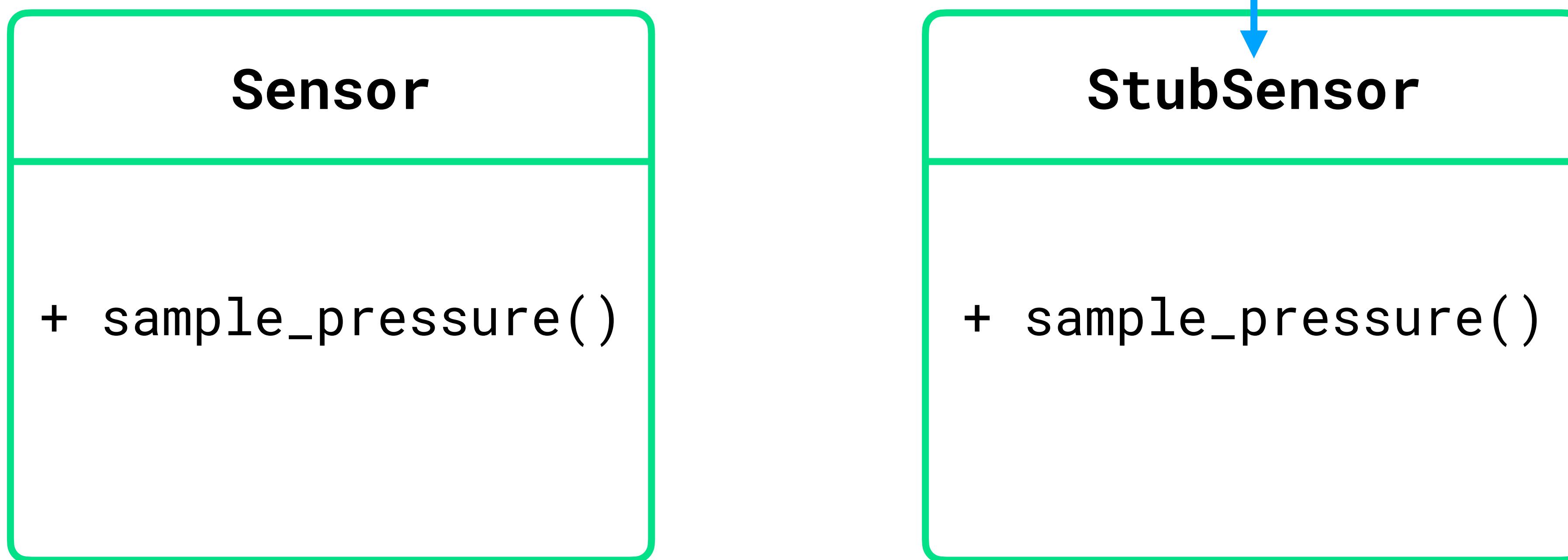
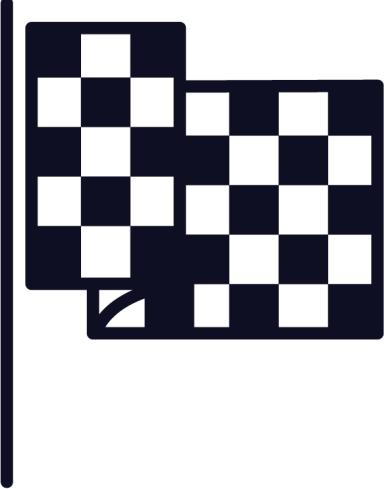
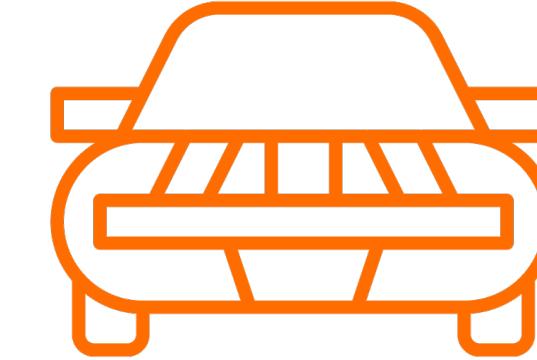
# Racing Car Example



Want to test  
this method

Replace this  
collaborator with  
a test double

# Racing Car Example

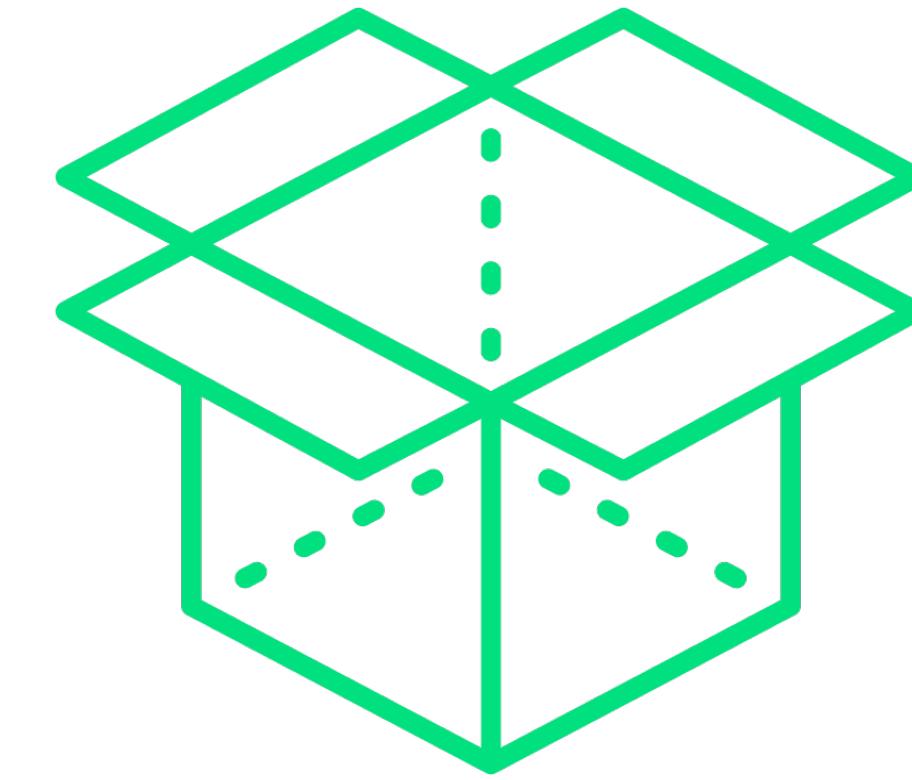


A **Stub** has the same methods as the class it replaces,  
but the implementation is very simple

# Stubs

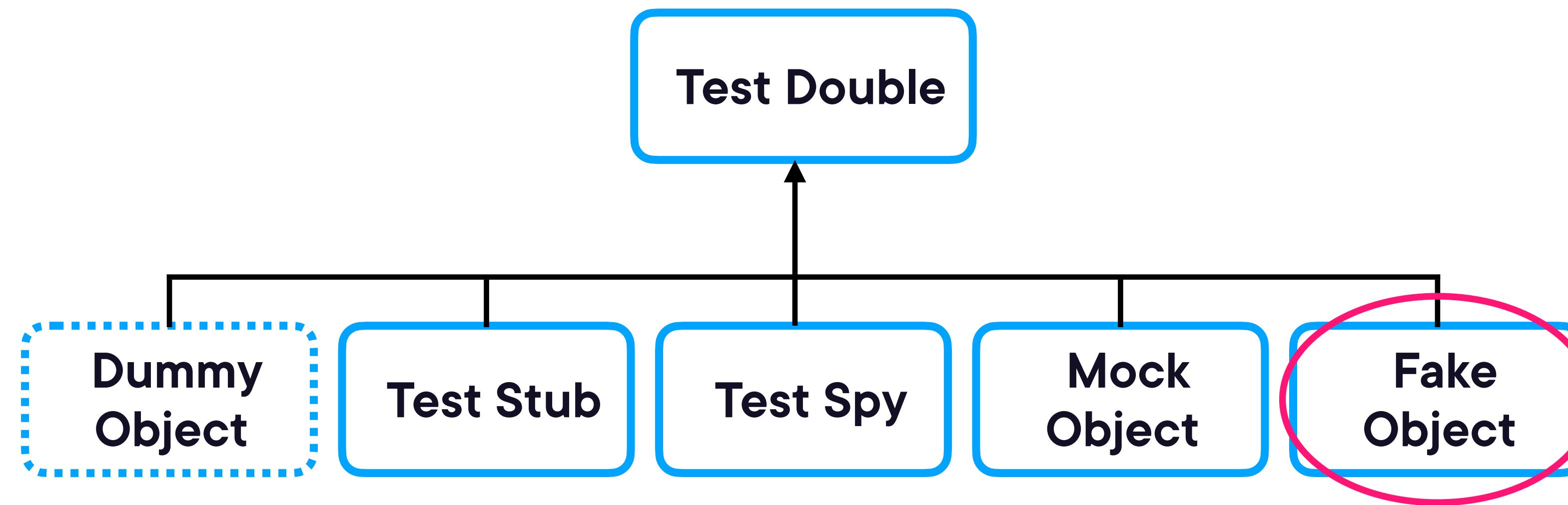


**Looks good from the outside**



**Contains nothing except  
what you put there**

# Different Kinds of Test Double - Fake

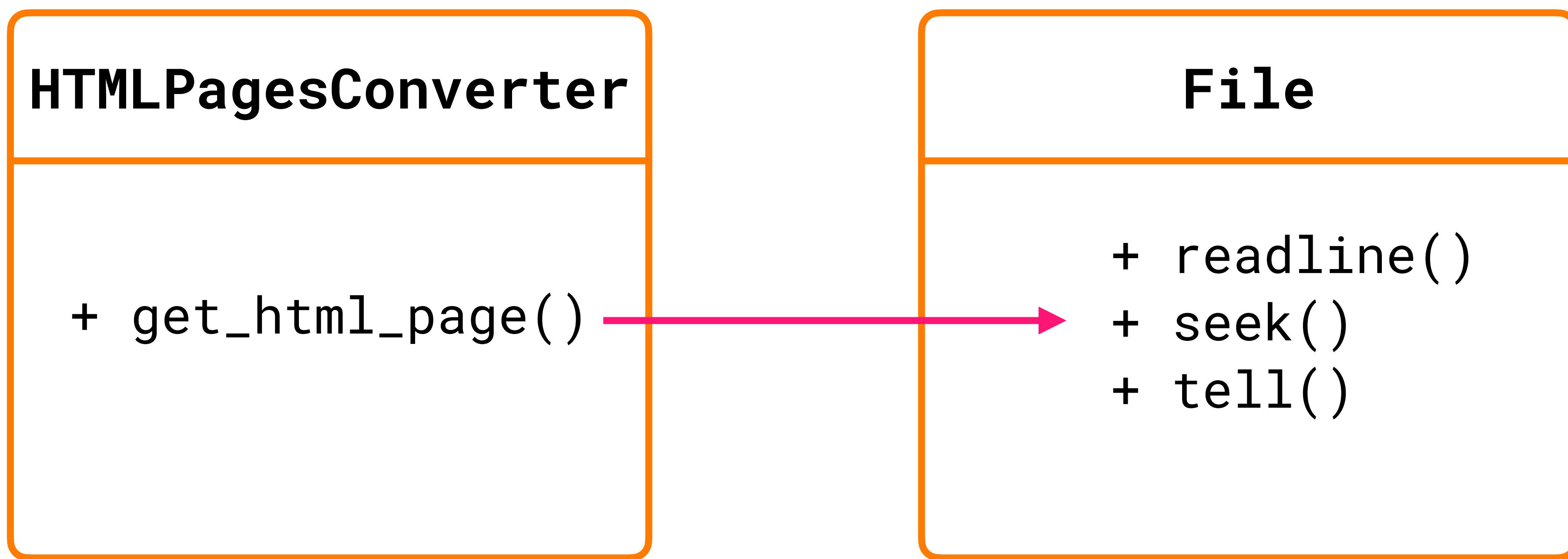
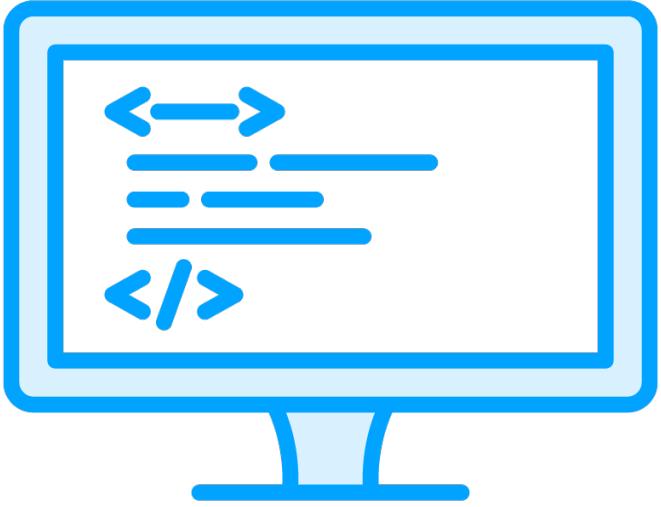


From <http://xunitpatterns.com>

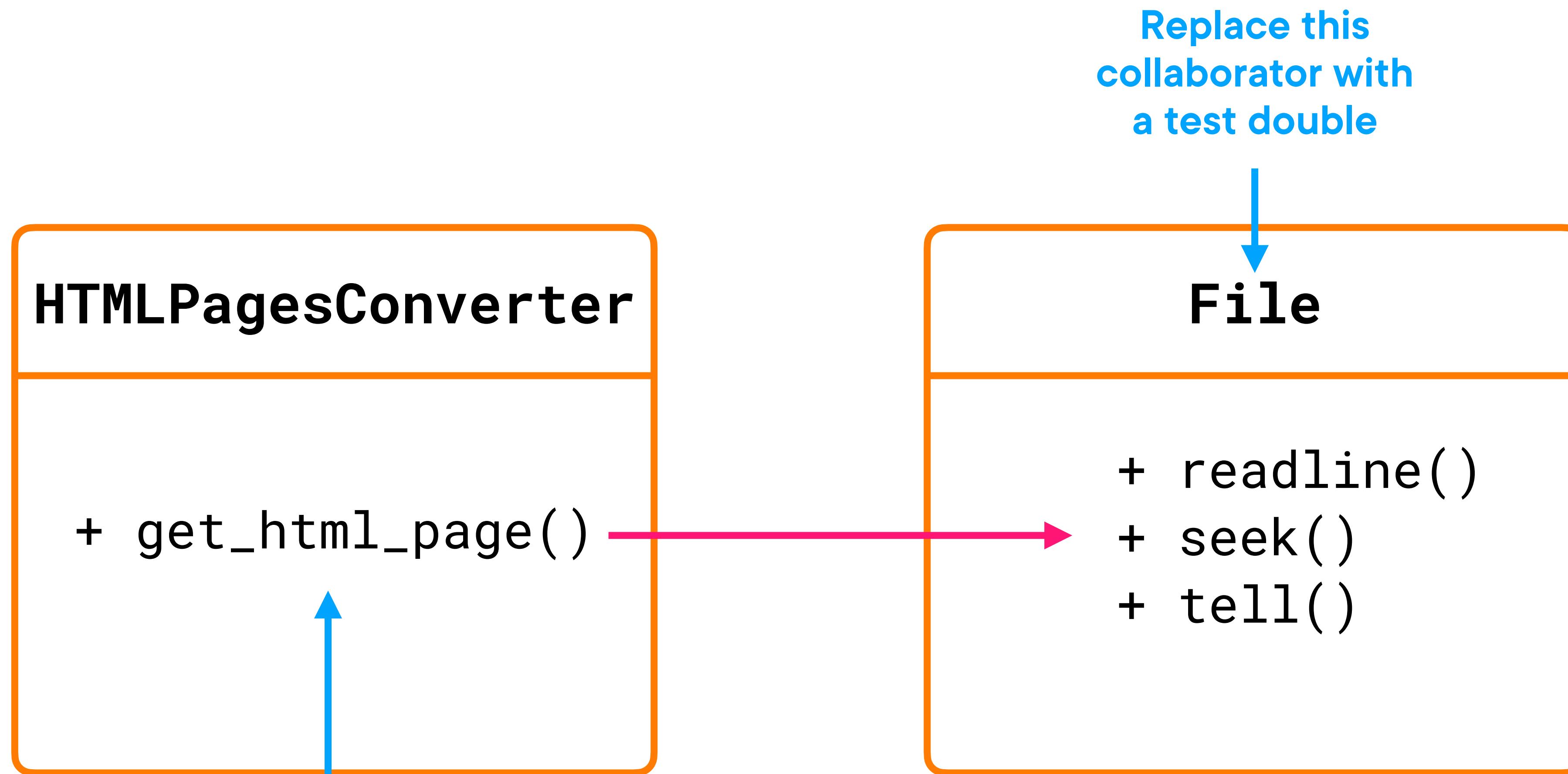
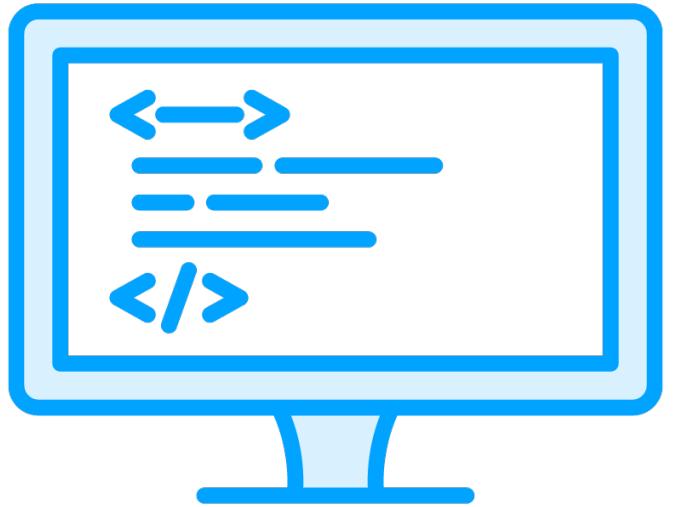
See also: [https://en.wikipedia.org/wiki/Test\\_Double](https://en.wikipedia.org/wiki/Test_Double)

## Using a Fake

# HTML Converter Example

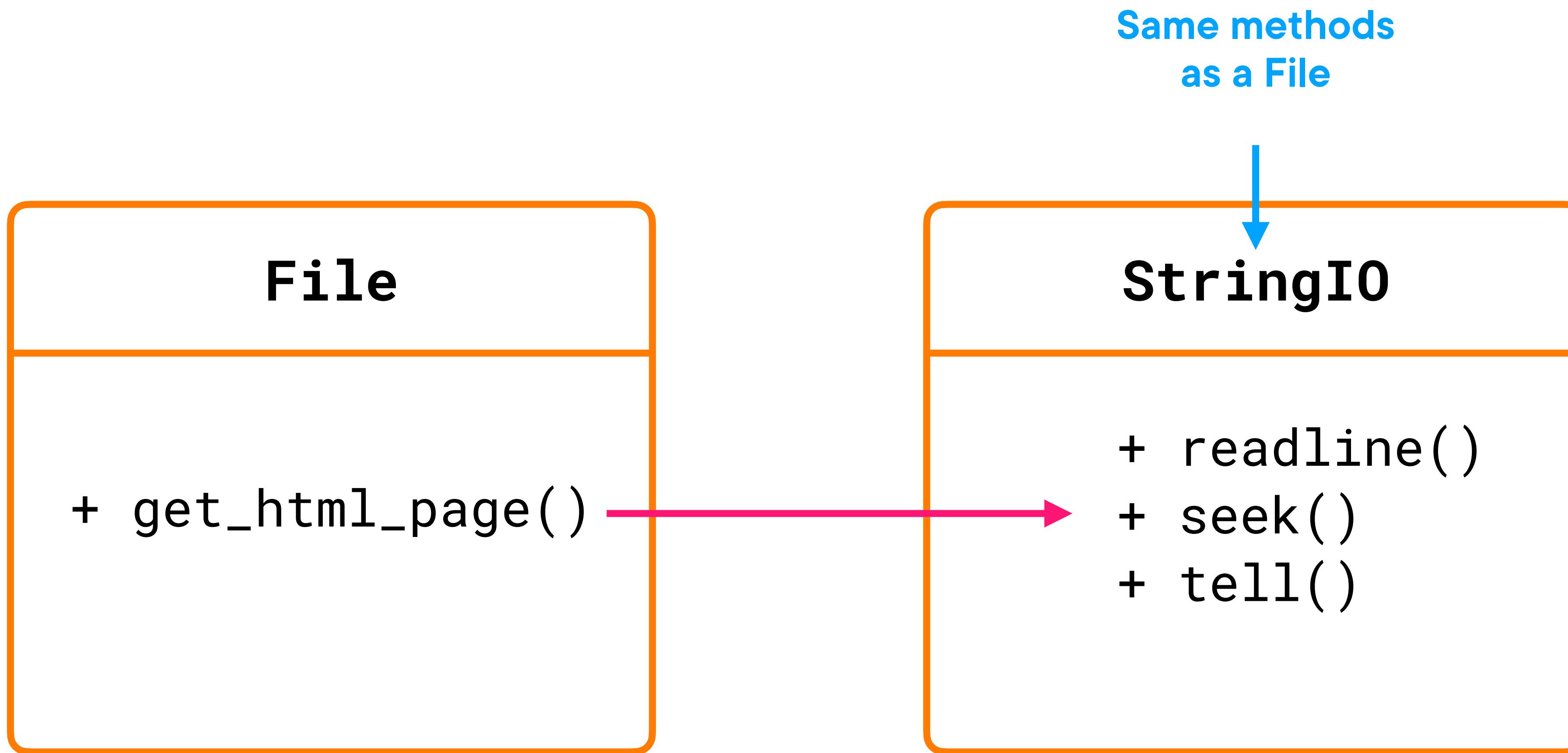
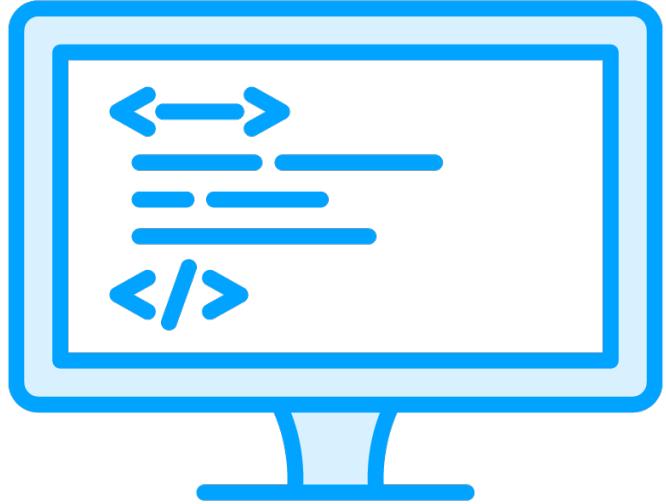


# HTML Converter Example



Want to test  
this method

# HTML Converter Example

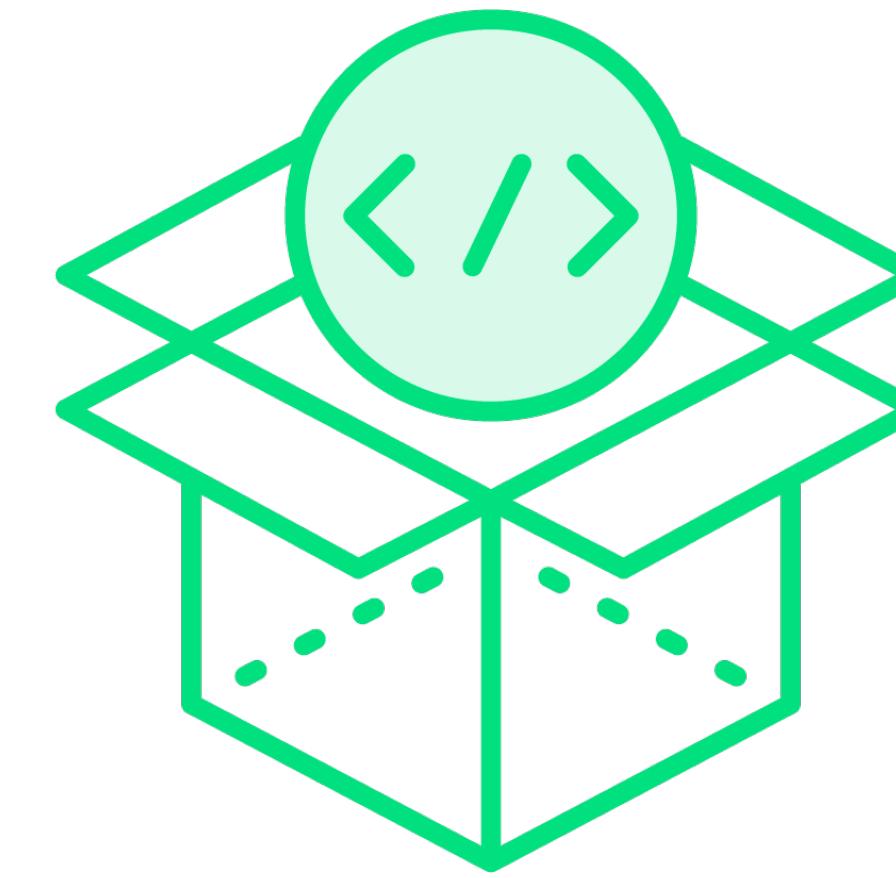


A Fake has an implementation with logic  
and behavior, but is not suitable for production.

# Fakes

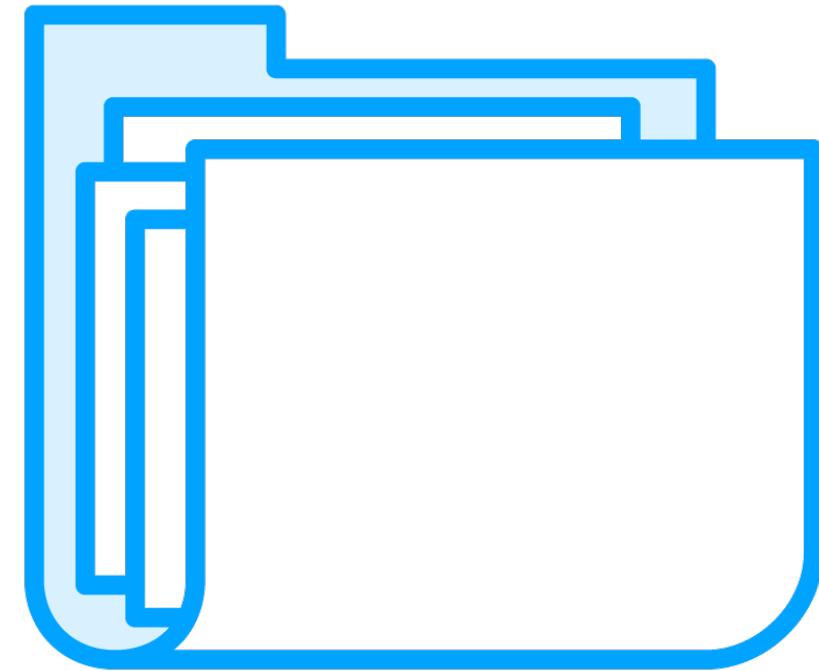


**Looks good from the outside**

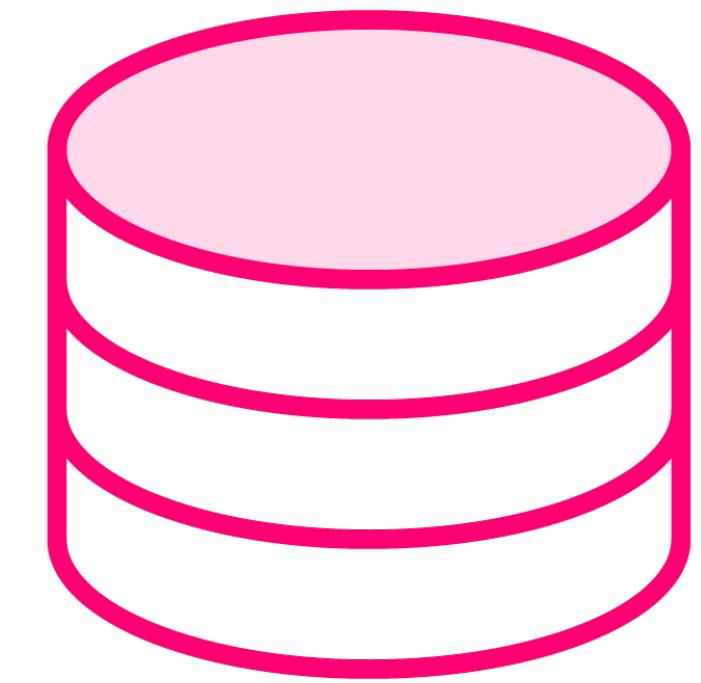


**Has an implementation  
with logic and behavior**

# Common Things to Replace with a Fake



**File**  
Replace with `StringIO`

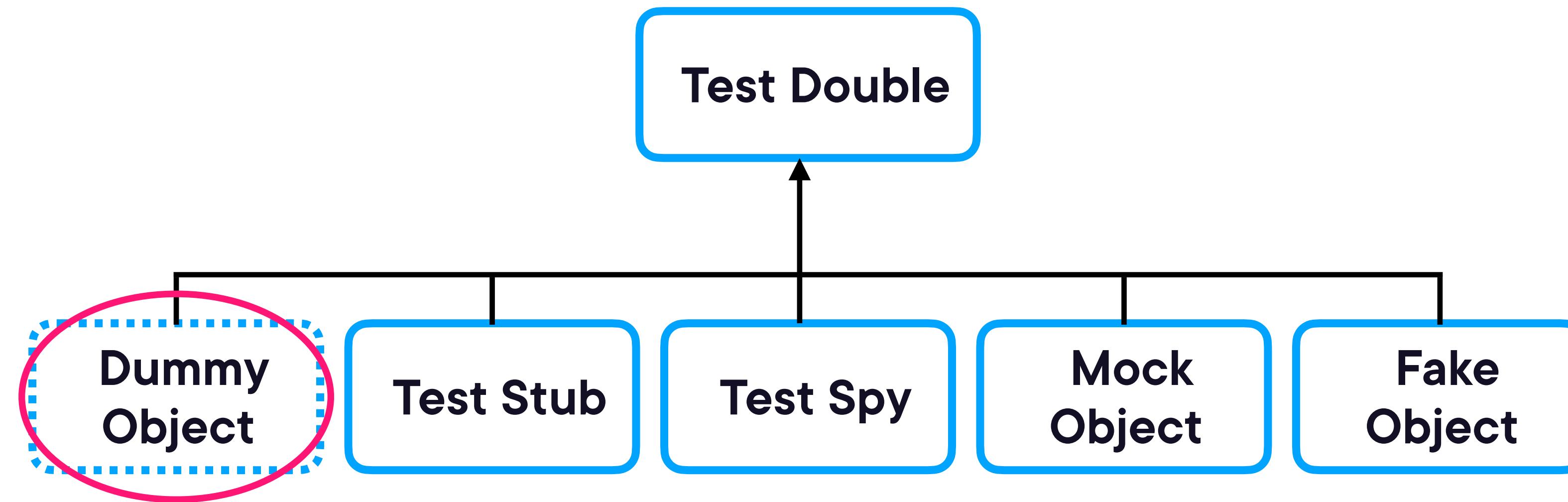


**Database**  
Replace with  
`in-memory database`



**Webserver**  
Replace with  
`lightweight Web Server`

# Different Kinds of Test Double - Dummy Object



From <http://xunitpatterns.com>

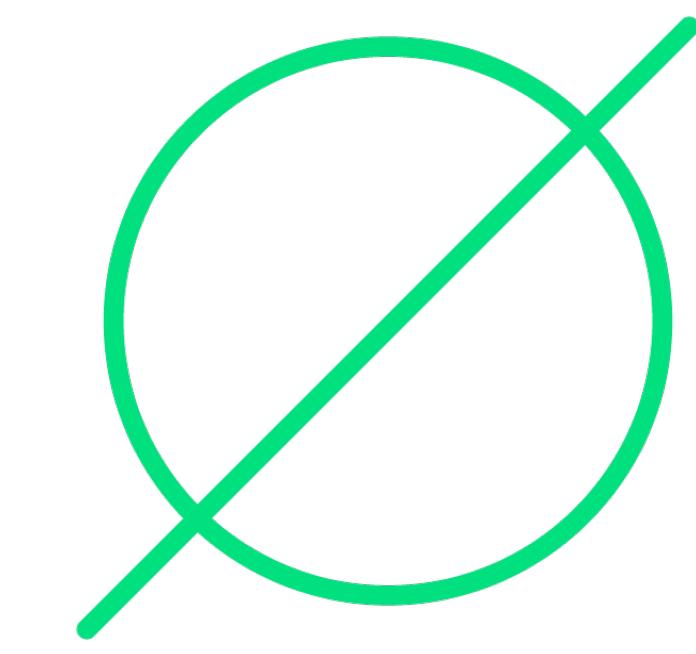
See also: [https://en.wikipedia.org/wiki/Test\\_Double](https://en.wikipedia.org/wiki/Test_Double)

## Using a Dummy

# Dummies

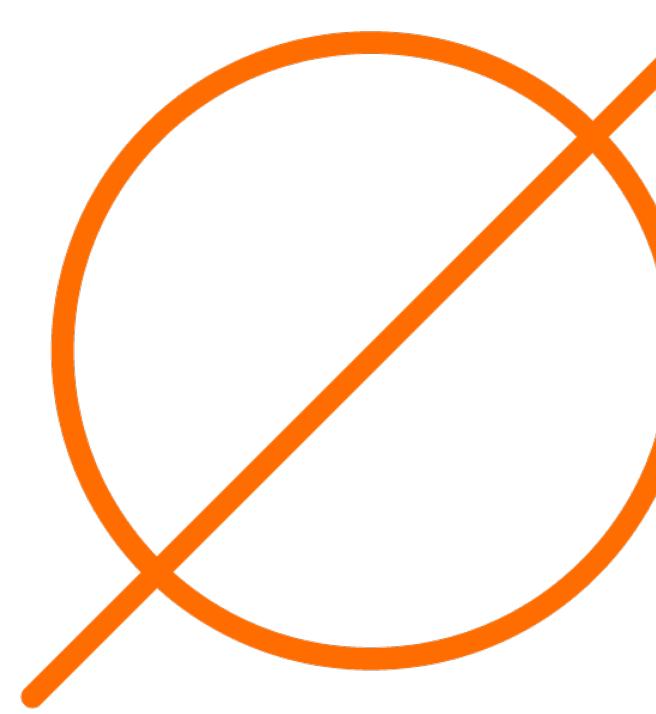


**It doesn't matter  
what it looks like**

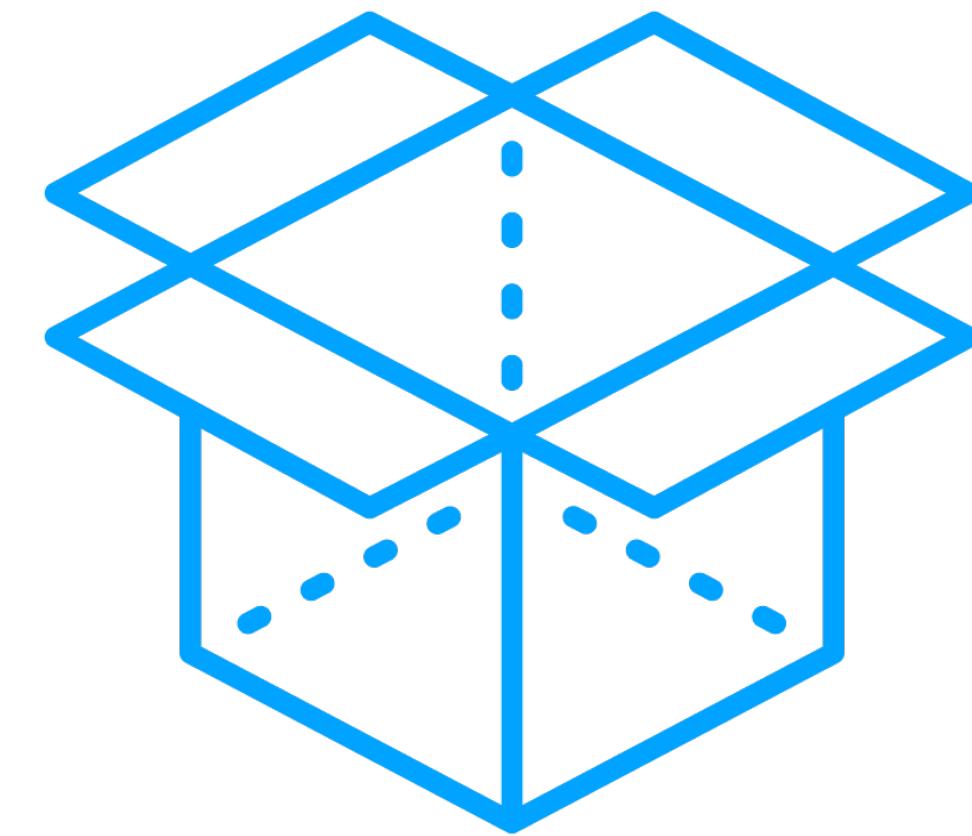


**A dummy is not used**

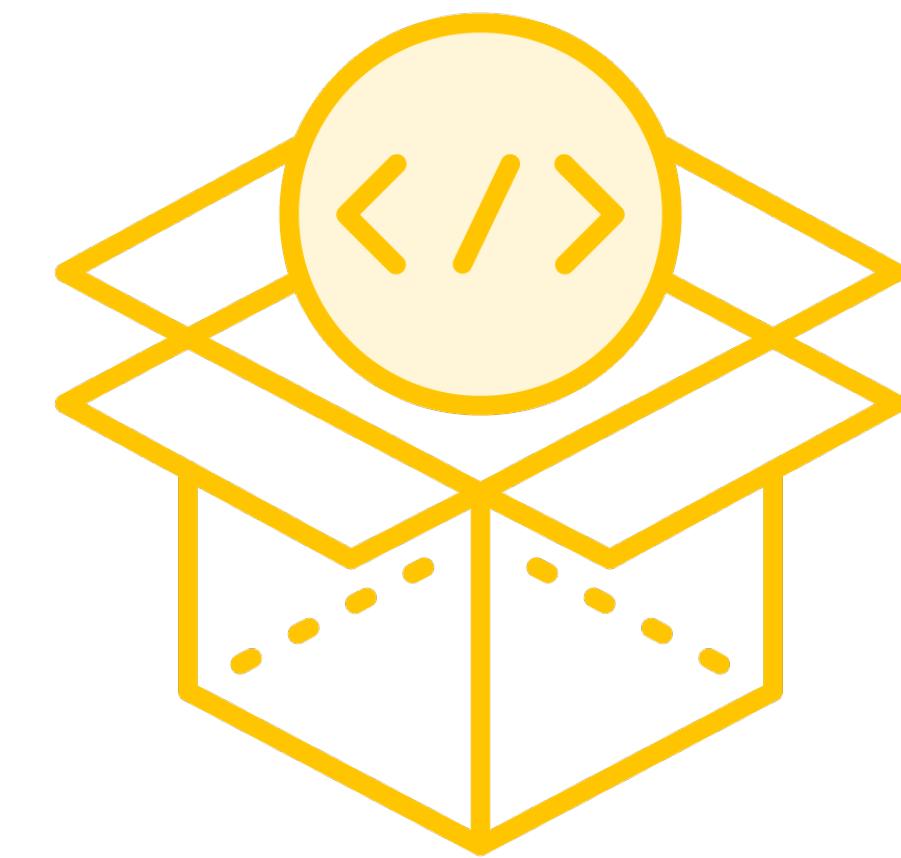
# Test Dummies Replace Collaborators



**Dummy**  
Usually None



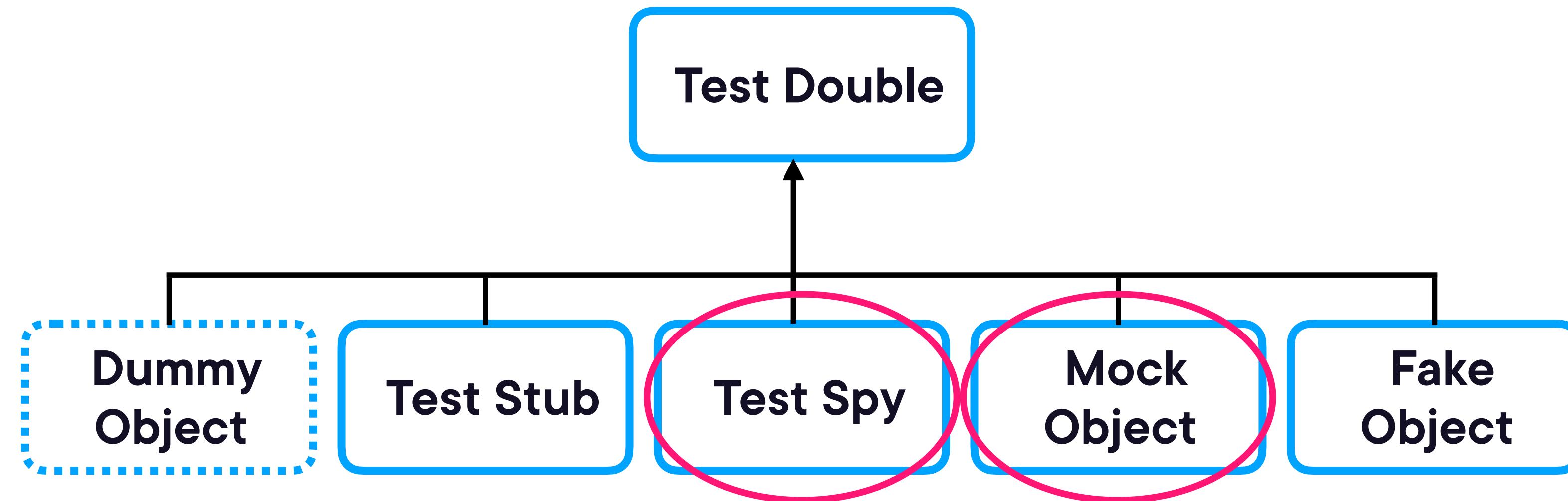
**Stub**  
Responds with fixed,  
pre-prepared data



**Fake**  
Has an implementation, but  
unsuitable for production

More similar to the collaborator being replaced

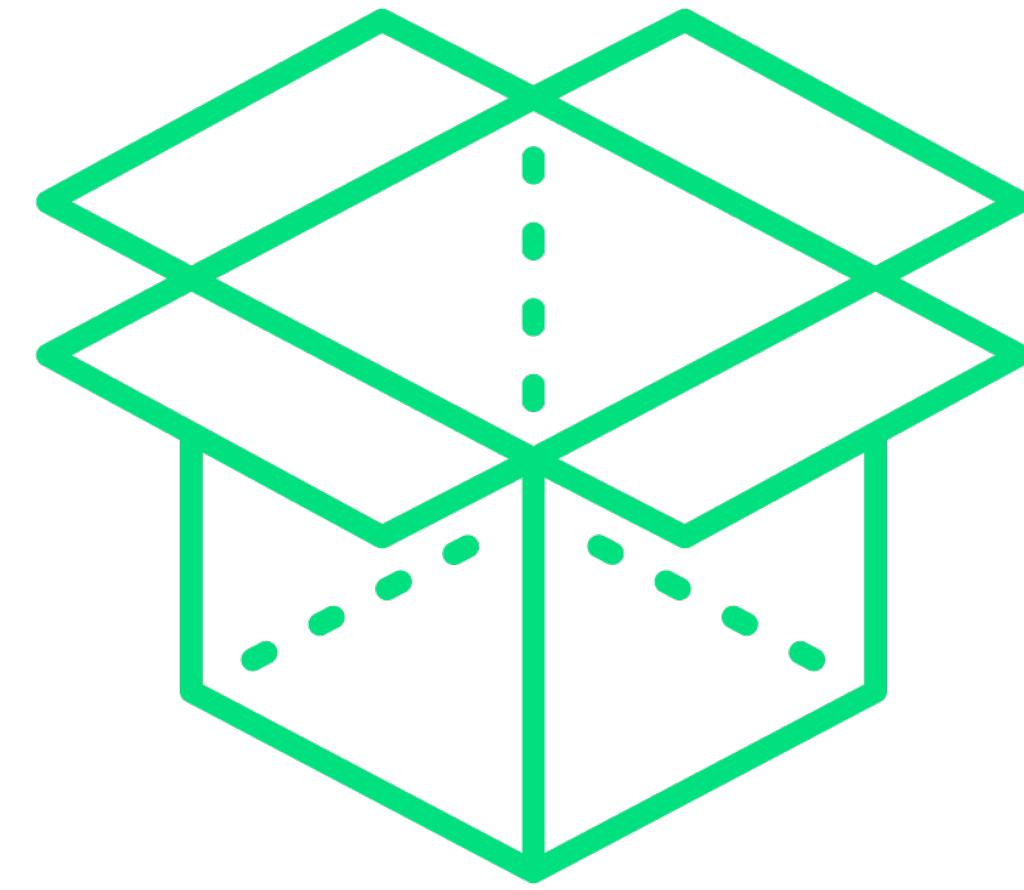
# Different Kinds of Test Double - Mocks and Spies



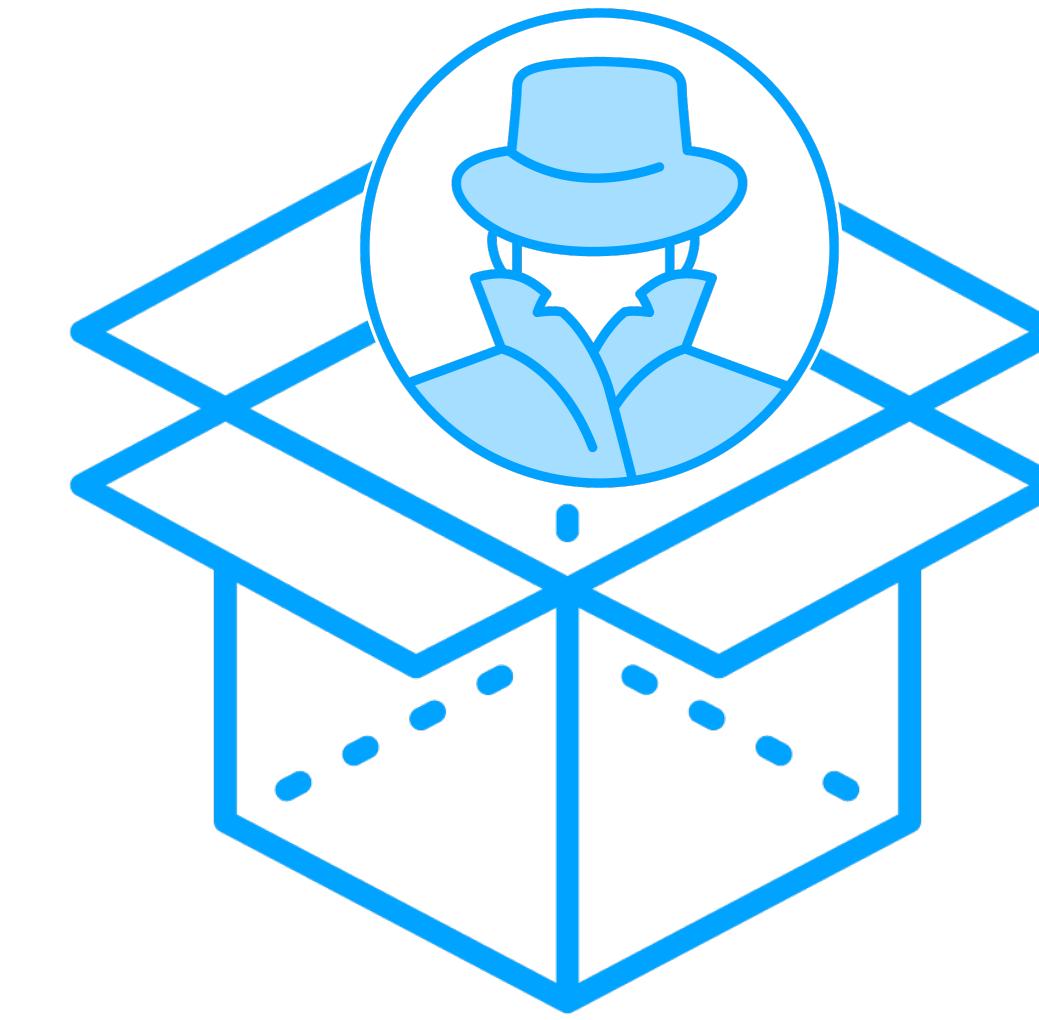
From <http://xunitpatterns.com>

See also: [https://en.wikipedia.org/wiki/Test\\_Double](https://en.wikipedia.org/wiki/Test_Double)

# Mocks and Spies Do More Than Stubs

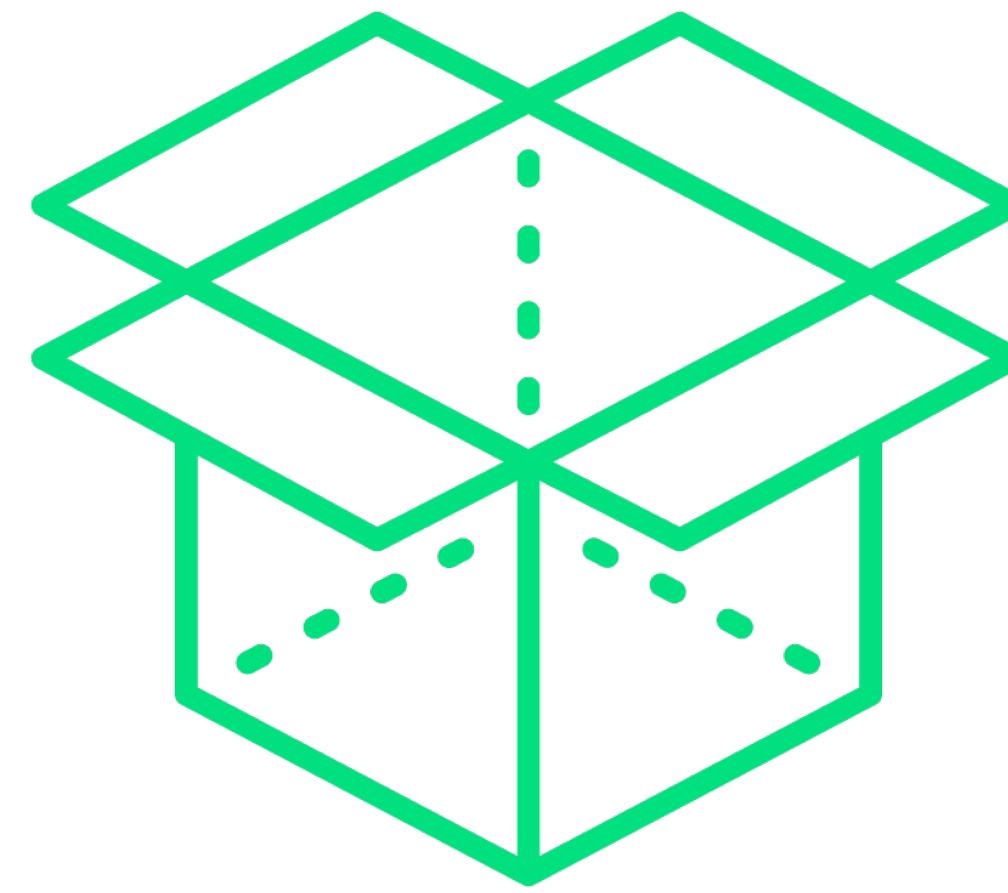


**Stub**  
Responds with fixed,  
pre-prepared data

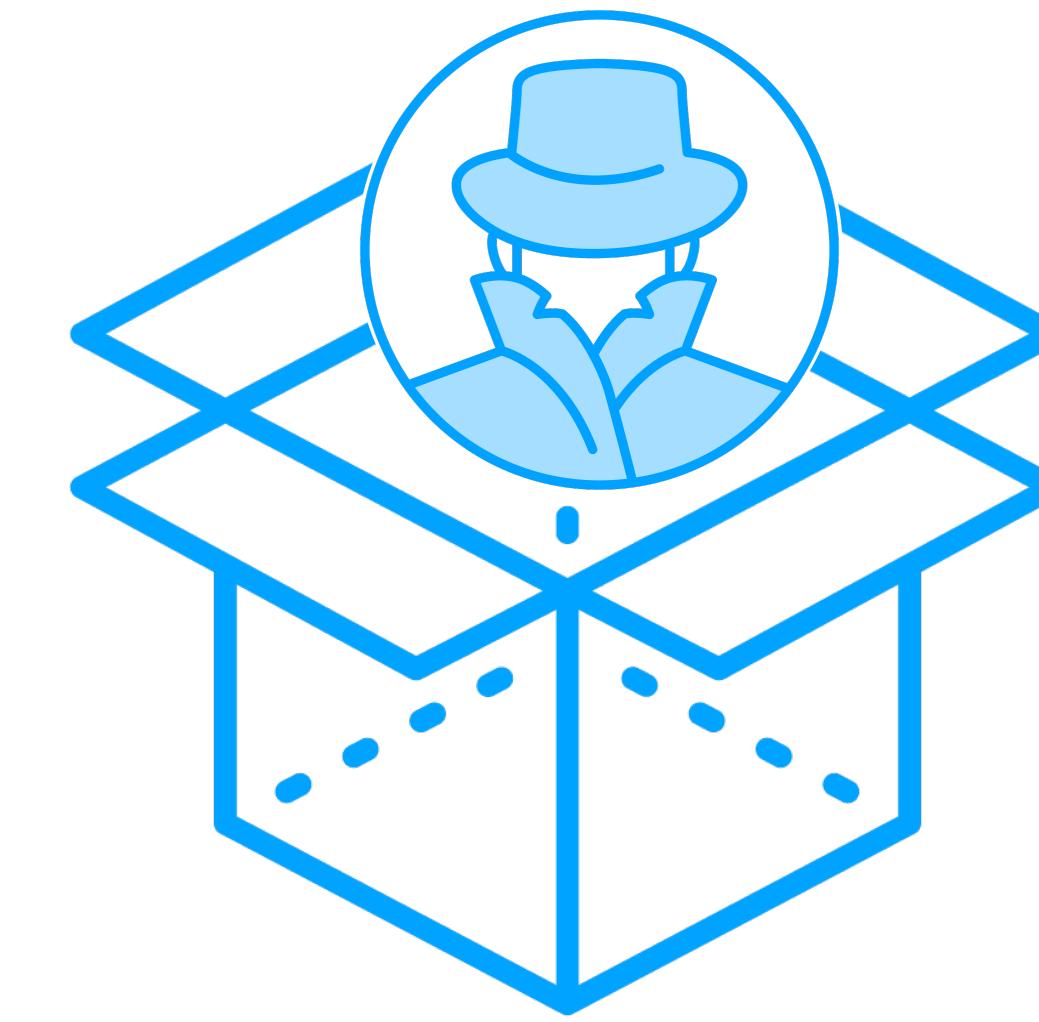


**Mock or Spy**  
Responds with fixed,  
pre-prepared data

# Mocks and Spies Do More Than Stubs

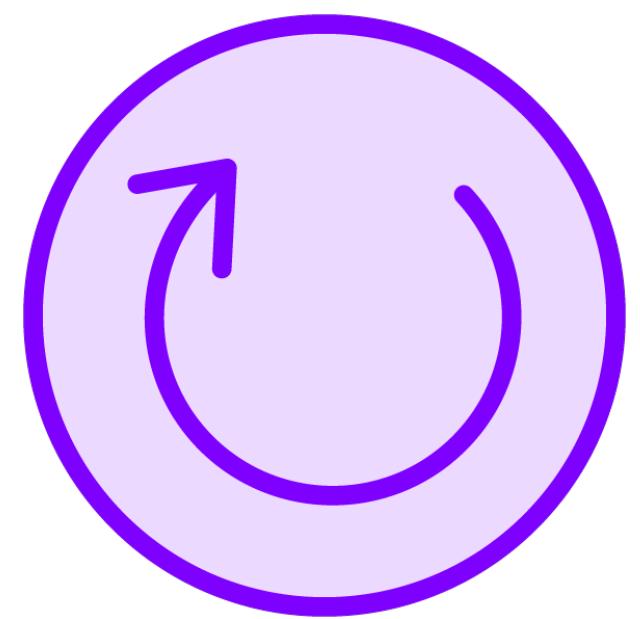


**Stub**  
Will not fail the test

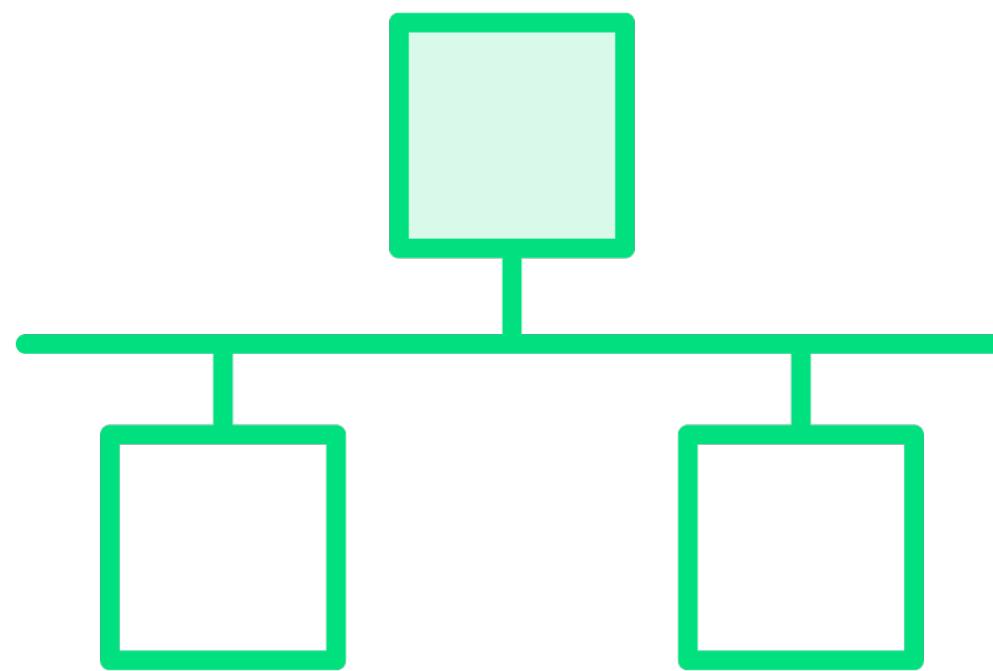


**Mock or Spy**  
Can fail the test if  
it's not called correctly

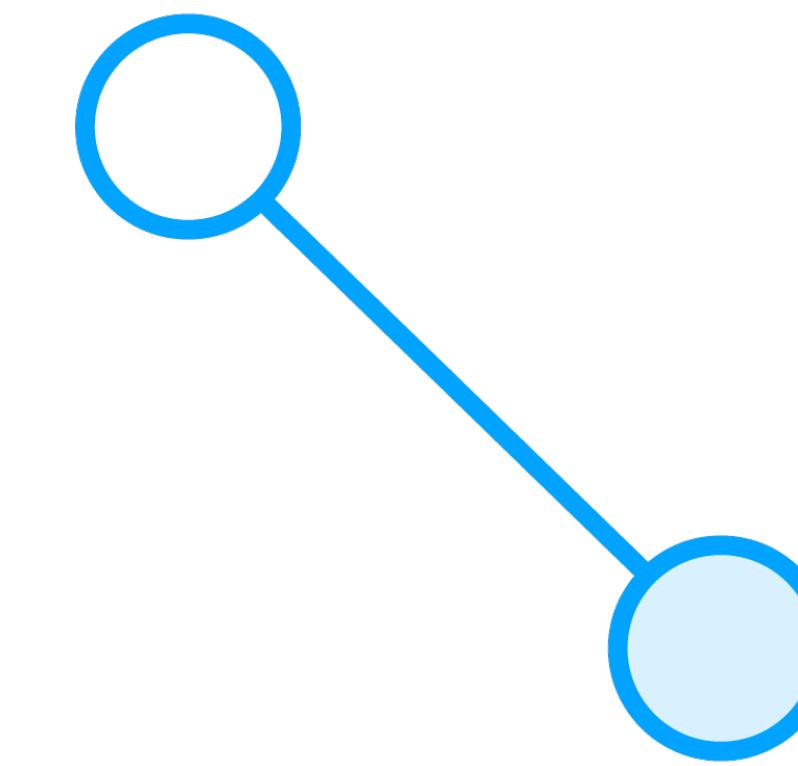
# Three Kinds of Assert



**Return Value**  
Or an exception



**State Change**  
Uses an API to query the  
new state



**Method Call**  
Check a specific function  
or method got called

Increasing Complexity

```
def test_convert_quotes():
    fake_file = io.StringIO("quote: ' ")
    converter = HtmlPagesConverter(open_file=fake_file)
    converted_text = converter.get_html_page(0)
    assert converted_text == "quote: &#x27;<br />"
```

## Assert on the Method Return Value

Add text here

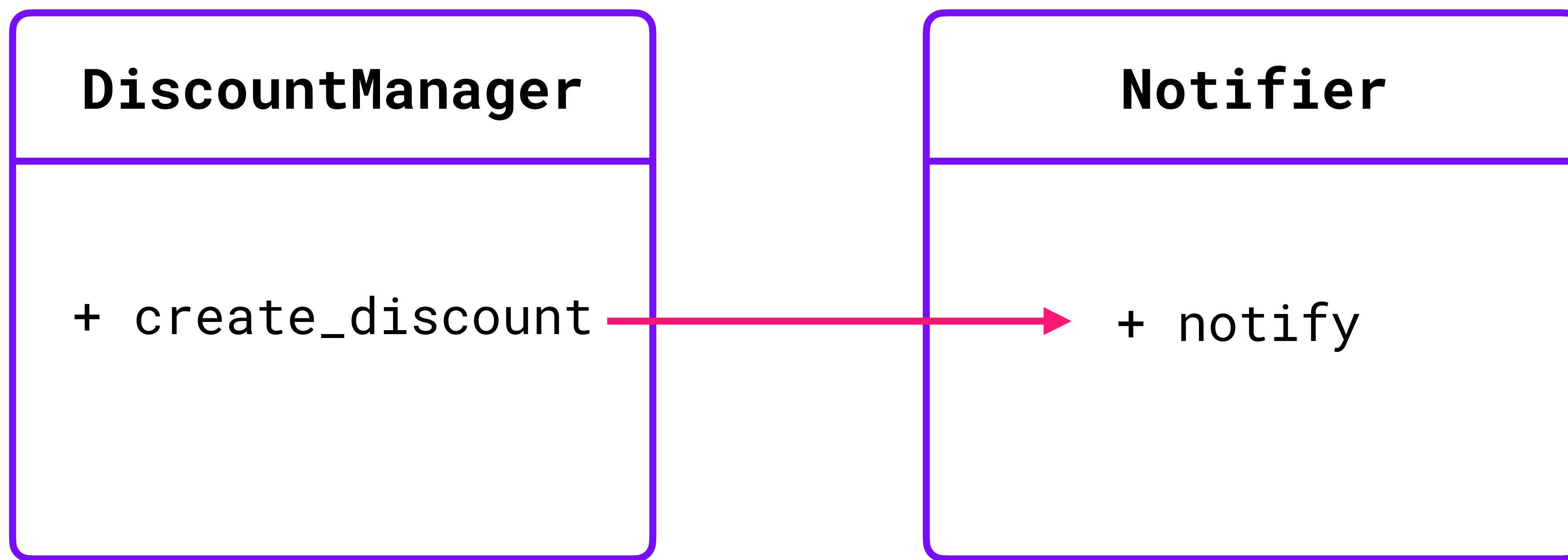
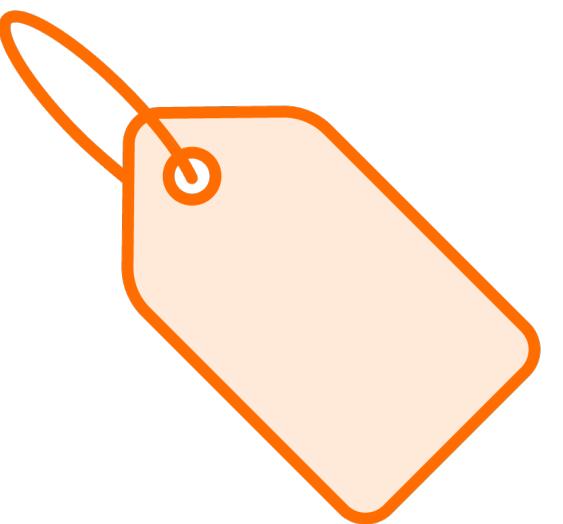
```
def test_alarm_is_off_by_default(self):  
    alarm = Alarm()  
    assert not alarm.is_alarm_on
```

## Assert on the Alarm State

Add text here

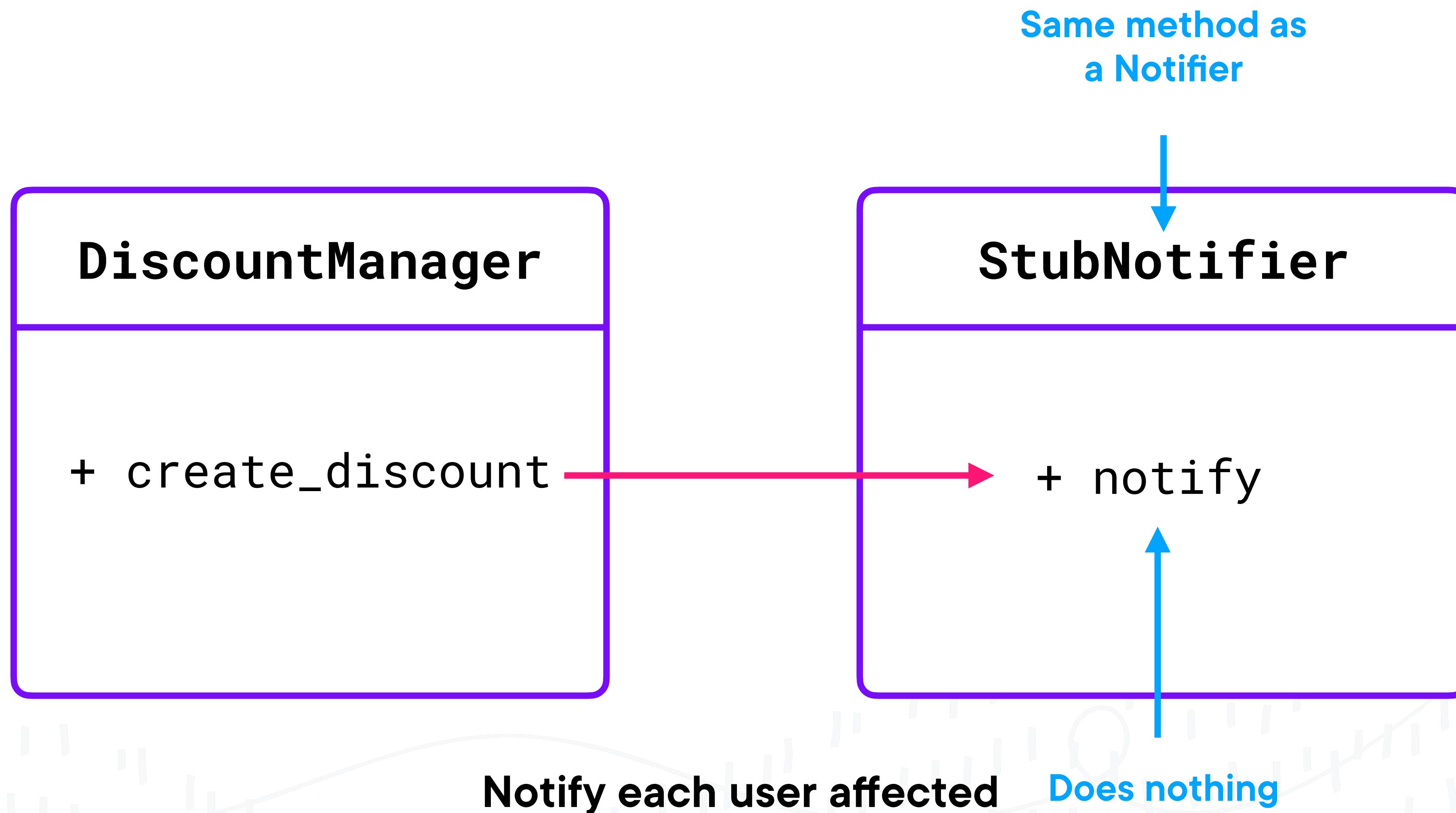
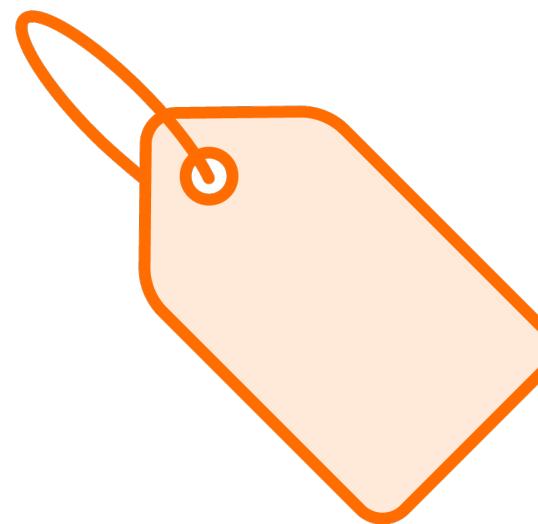
## Using a Spy

# Discount Manager Example

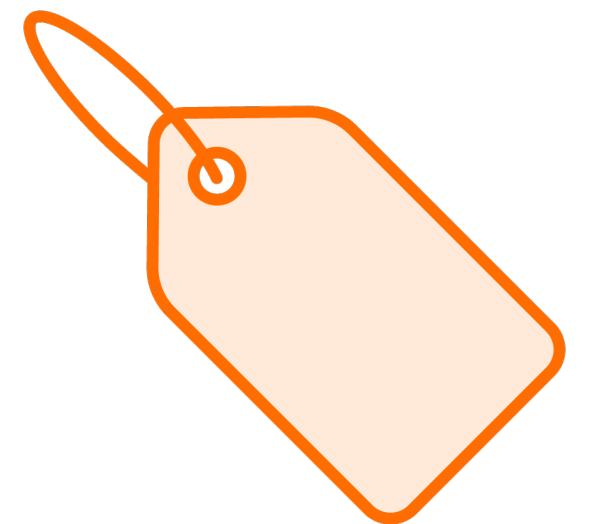


Notify each user who might be interested

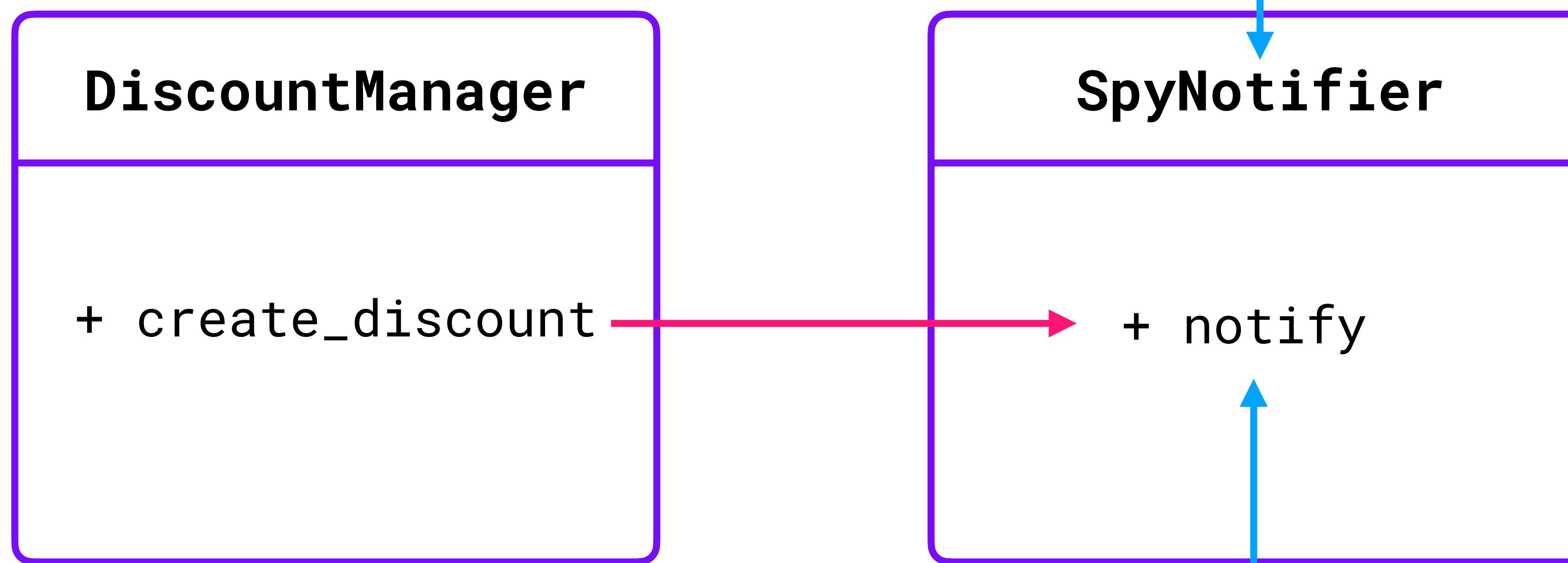
# Discount Manager Example



# Discount Manager Example



A Spy records the method calls it receives, so you can assert they were correct



Notify each user affected

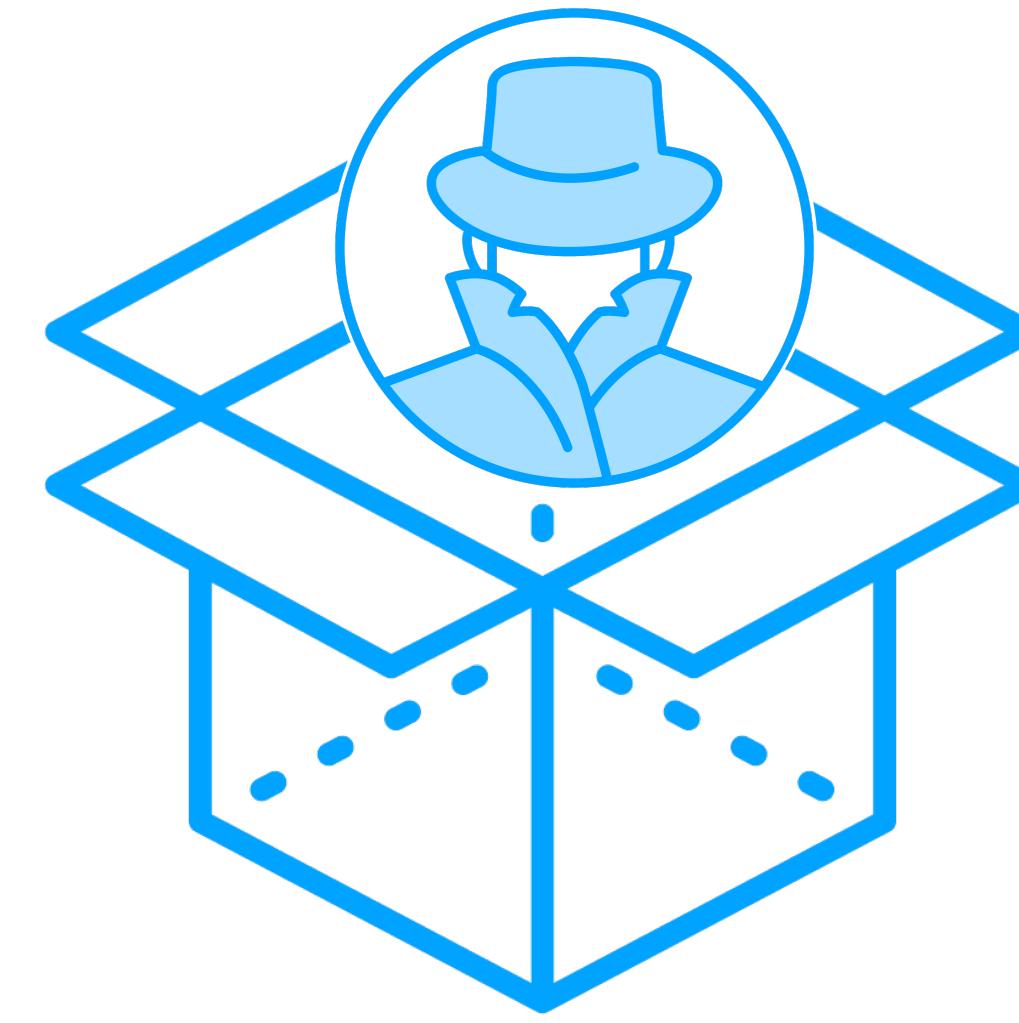
Records each call

Same method as a Notifier

# Spy

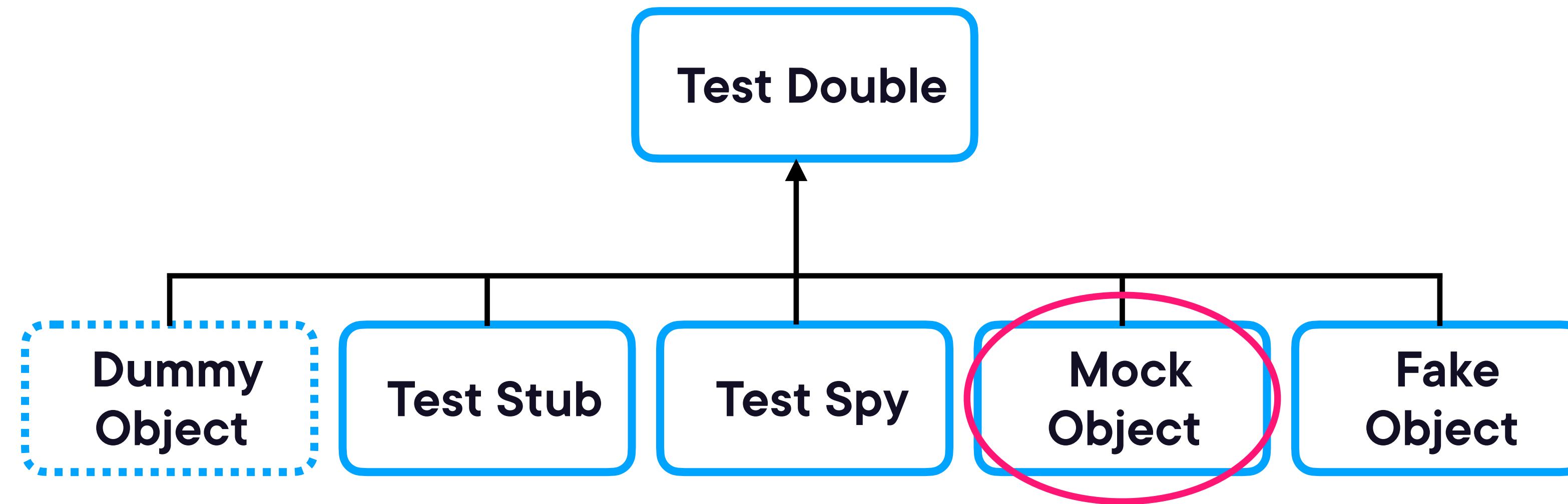


**Looks good  
from the outside**



**Records any interactions,  
so you can check them later**

# Different Kinds of Test Double - Mock Object

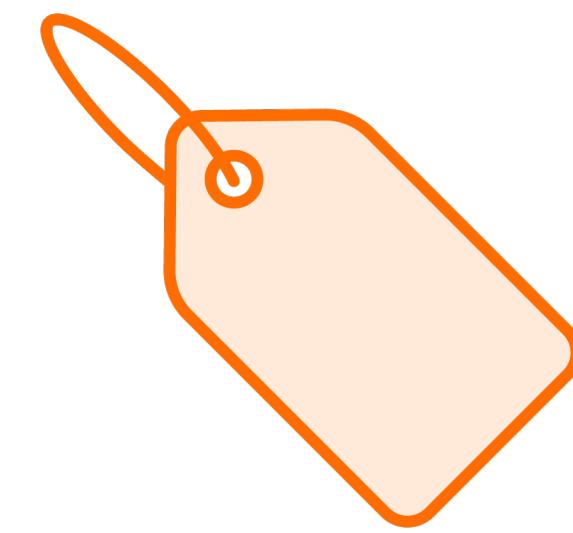


From <http://xunitpatterns.com>

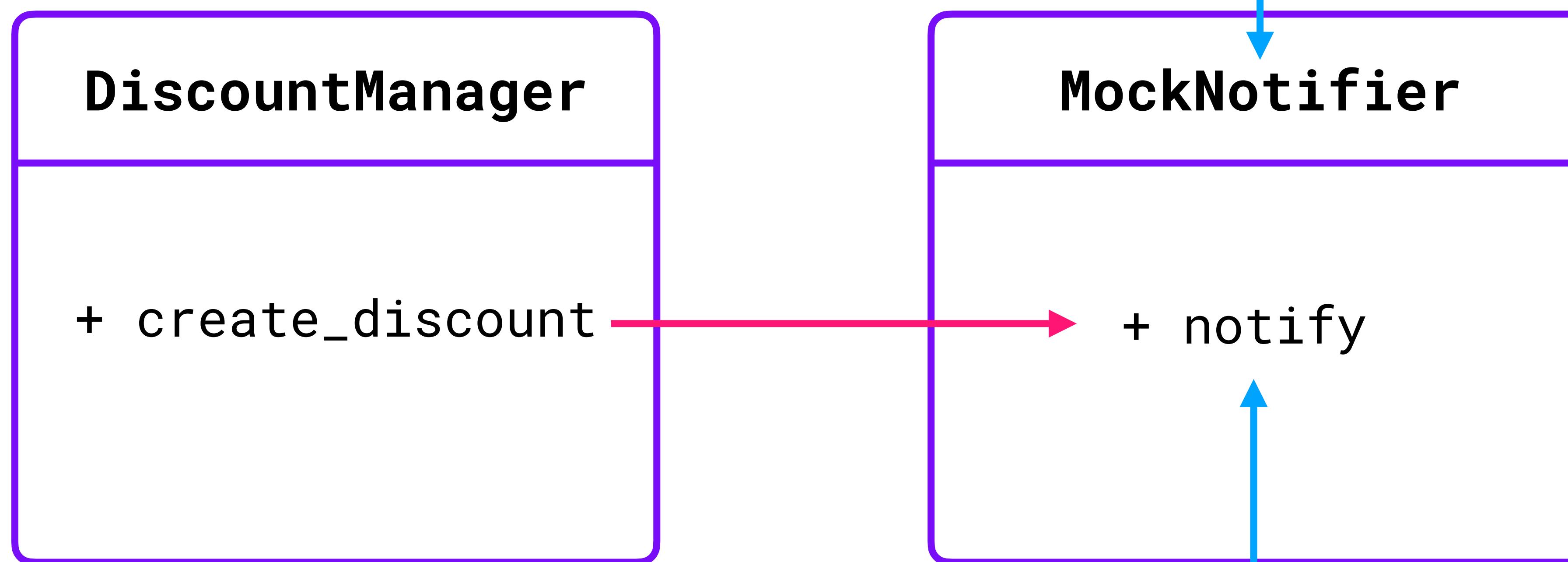
See also: [https://en.wikipedia.org/wiki/Test\\_Double](https://en.wikipedia.org/wiki/Test_Double)

## Using a Mock

# Discount Manager Example



A Mock knows what method calls to expect, and fails the test if they are incorrect.



Notify each user affected

Checks each  
call is correct

# Mocks



**Looks good  
from the outside**



**Expect certain method calls,  
otherwise raise an error**

# Spies and Mocks

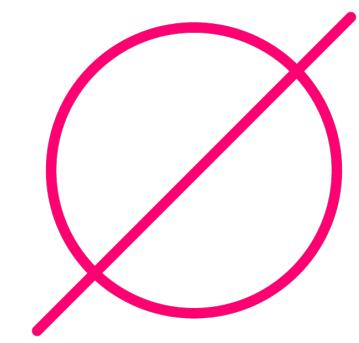


Spy

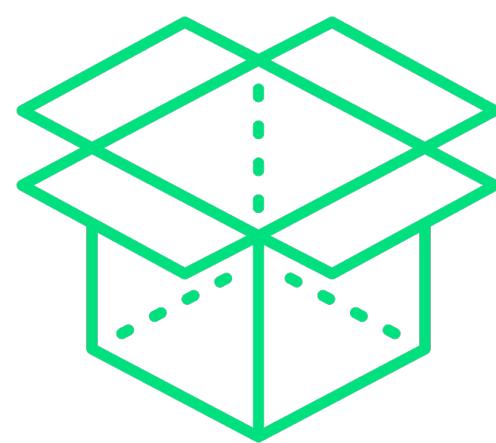


Mock

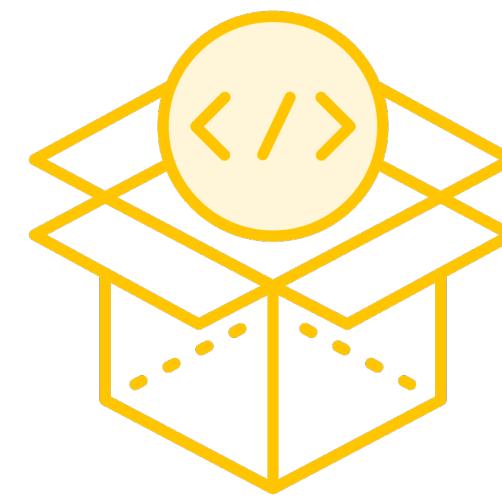
# Test Doubles - How to Use Them



Dummy

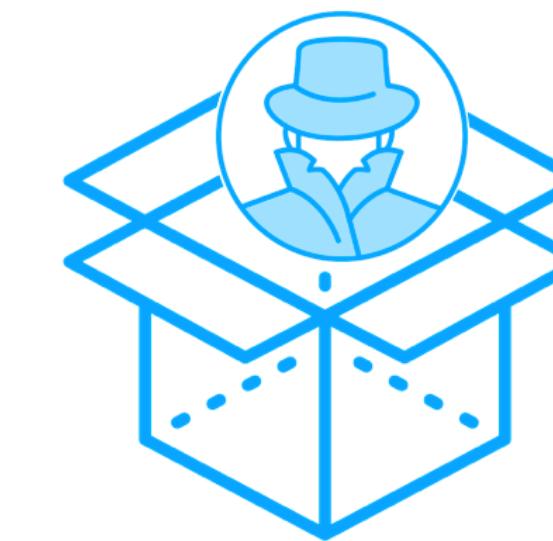


Stub



Fake

⋮



Spy



Mock

Replace an awkward collaborator

Ensure interactions are correct

# Summary

## Kinds of Test Double:

- Stubs
- Fakes
- Dummies
- Spies
- Mocks