Module 3.1:
"Null Object"

TEKNOLOGISK
INSTITUT

Agenda

- Introductory Example: Animal Sounds
- Challenges
- Pattern: Null Object
- Implementing the Null Object Pattern
- Overview of Null Object
- Null Object in Unit Testing

Introductory Example: Animals Sounds

```
class AnimalFactory : IAnimalFactory
{
    public IAnimal Create( string description )
    {
        ...

        if (_animalTypes.TryGetValue(processedDescription,
             out Type animalType))
        {
            return Activator.CreateInstance(animalType) as IAnimal;
        }
        return null;
    }
    ...
}
```

```
interface IAnimal
{
    string Name { get; }
    void MakeSound();
}
```

Challenges

- C# has specialized syntax for null-checks, but could we relieve the client of that burden?

- What if a component needs an object to compile and run, but during unit tests that object should be "inactive"?

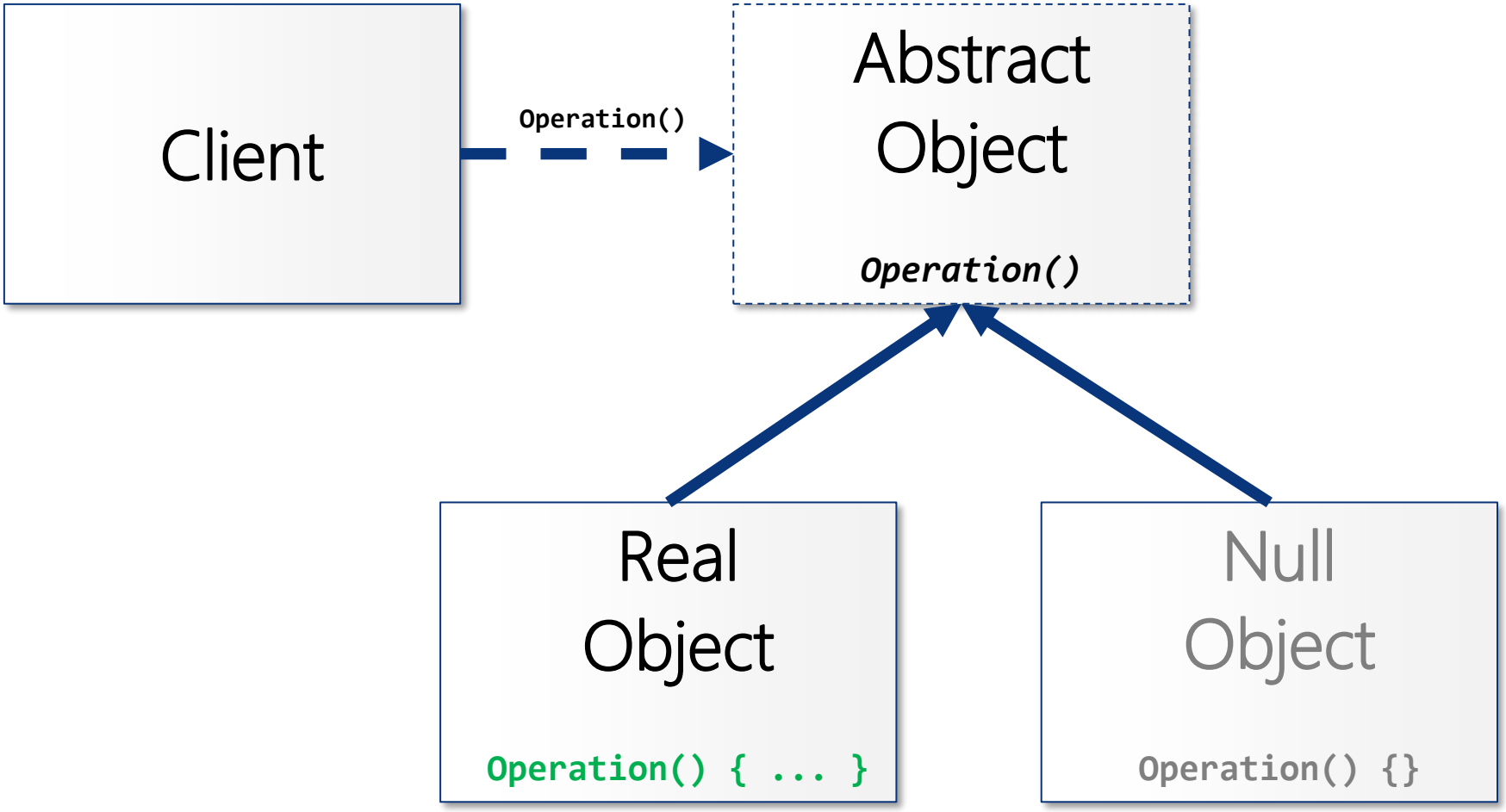
Pattern: Null Object

- *Provide an object as a surrogate for the lack of an object of a given type. The Null Object provides intelligent "do-nothing" behavior, hiding the details from its collaborators.*

- Outline
  - Abstract the handling of null references away from the client
  - Create an object with do-nothing behavior in a well-defined interface expected by the client

- Origin: Bobby Woolf (1998)

Overview of Null Object Pattern



Overview of Null Object Pattern

- Client
  - Needs a collaborator exposing `Operation()`
- Abstract Object
  - Interface or abstract class specifying the abstract `Operation()`
- Real Object
  - Concrete class implementing the Abstract Object interface
  - Supplied appropriate behavior in `Operation()` used by Client
- Null Object
  - Concrete class implementing the Abstract Object interface
  - Can be substituted for Real Object in the context of Client
  - Implements the `Operation()` to do nothing / neutral behavior
  - The exact neutral behavior depends on what Client expects

Null Object in Unit Testing

- Null Objects are extremely useful in unit testing
  - "mock", "stub", ...

```
private class NullLogger : ILogger
{
    public void Enter( string callerMemberName ) { }
    public void Error( string message ) { }
    public void Error( Exception exception ) { }
    public void Exit( string callerMemberName = null ) { }
    public void Info( string message ) { }
    public void Info( Exception exception ) { }
}
```

- Null objects and factories can be set up in `[TestInitialize]`


WINCUBATE
Jesper Gulmann Henriksen
PhD, MCT, MCSD, MCPD
Phone : +45 22 12 36 31
Email : jgh@wincubate.net
WWW : http://www.wincubate.net
Ringgårdsvej 4A
8270 Højbjerg
Denmark