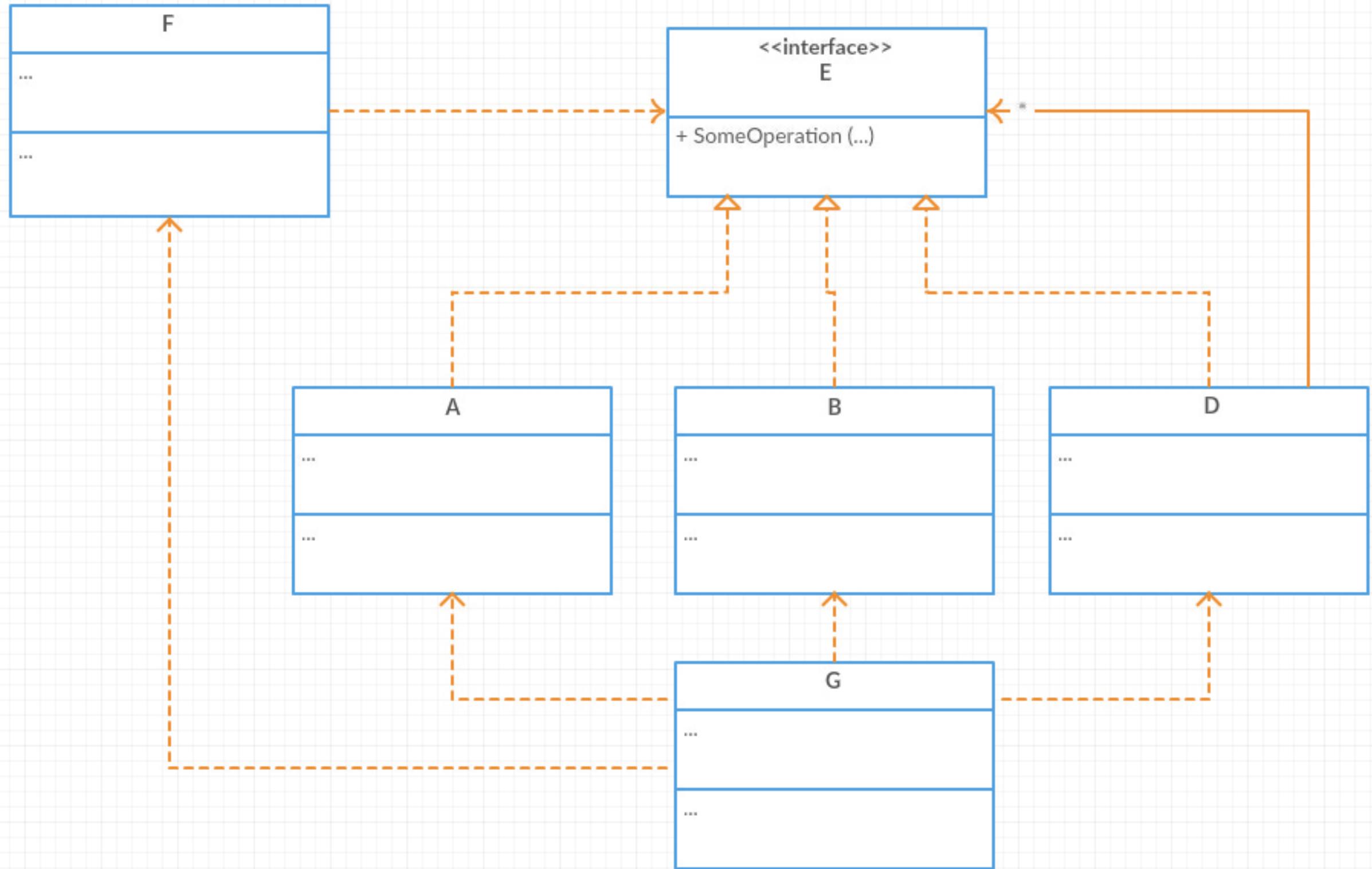


## Question 1

The class diagram below shows a design that use a pattern. Which pattern is this?



Wrong Answer!

Your answer was:

- Observer
- Strategy
- Visitor
- Facade
- Composite

## Question 2

The pseudo code below is badly designed because it has not considered a pattern which?

```
class A {
    public void SomeOperation() {
        ...
    }
    public bool SomeCheck() {
        ...
    }
}

class B {
    private list<A> m_as;

    public Iterable<A> Get() {
        return m_as;
    }
}

class C {
    B m_b;

    public void PerformSomeOperationOnCheck() {
        foreach (A a in m_b.Get()) {
            if (a.Check()) {
                a.SomeOperation();
            }
        }
    }

    public void PerformSomeOperationOnNOTCheck() {
        foreach (A a in m_b.Get()) {
            if (a.Check() != true) {
                a.SomeOperation();
            }
        }
    }
}
```

Wrong Answer!

Your answer was:

- Visitor
- Information Expert
- Observer
- Facade
- Polymorphism
- Strategy

### Question 3

The domain model

**Wrong Answer!**

Your answer was:

- shows dependencies between classes
- should be understandable by an end-user/customer stakeholder
- focus on dynamic behavior and rules
- shows navigability
- includes operations in classes
- shows software concepts
- shows associations between classes

## Question 4

What is a Design Pattern

Wrong Answer!

Your answer was:

- A named, abstract problem and abstract solution.
- A way to restructure your code without changing the functionality.
- A specific way of solving problems in object oriented design.

## Question 5

Given the following pseudo code for a class definition how should the relation between A and B be modeled in UML?

### Correct Answer!

Your answer was:



## Question 6

What differentiates a Design model from a Domain Model

### Wrong Answer!

Your answer was:

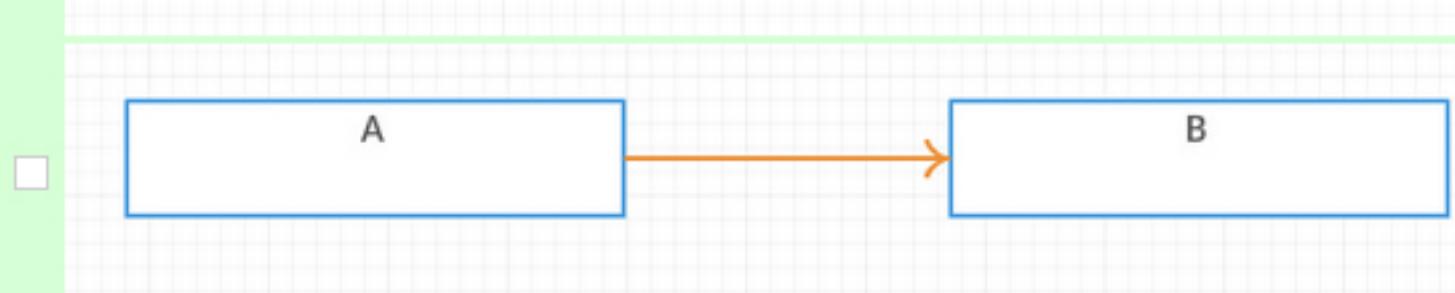
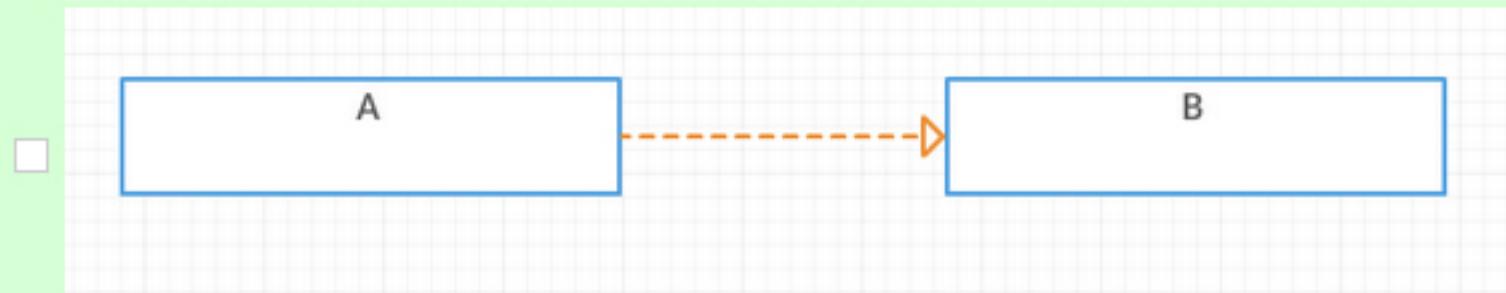
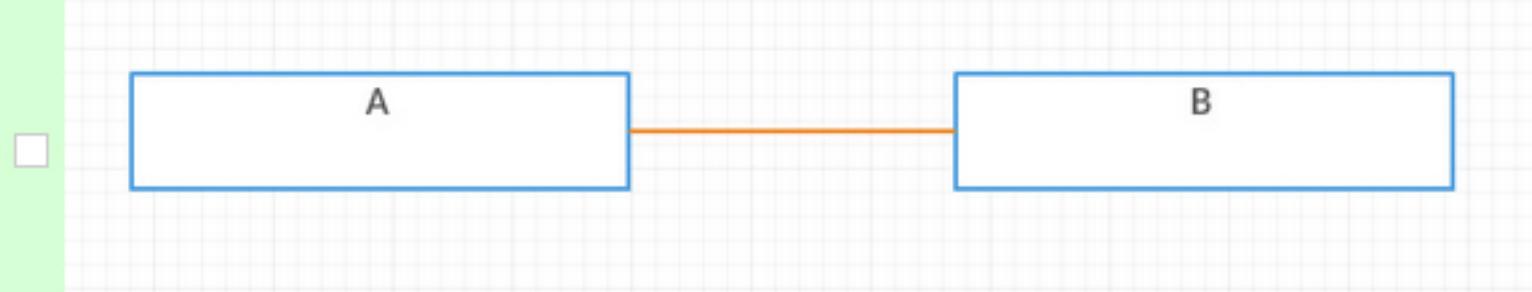
- The Design model uses class diagrams the Domain model does not
- The Domain Model models Reality and Not Software
- The Design Model is an Analysis Artefact
- The Domain Model is an Analysis Artefact
- The Design Model models Reality and Not Software

## Question 7

Which of the following class diagrams displays a Generalization/Specialization relation between A and B?

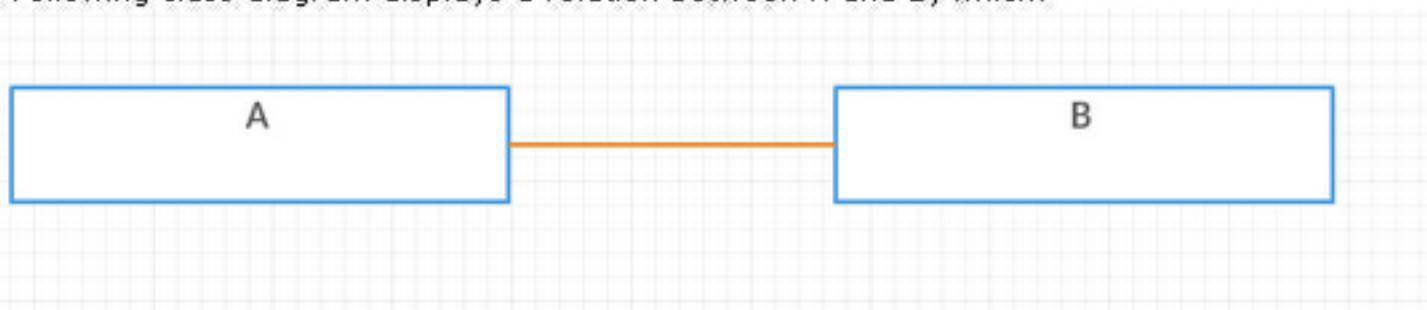
**Correct Answer!**

Your answer was:



## Question 8

The Following class diagram displays a relation between A and B, which?



**Correct Answer!**

Your answer was:

- Realization
- Dependency
- Association
- Generalization/Specialization

## Question 9

Two operations in the same class have some duplicated code in them, what should you do?

### Wrong Answer!

Your answer was:

- Use the Strategy Pattern
- Create a private operation, move the duplicated code there and add the necessary parameters.
- Create an abstract base class and move the duplicated code to this class, then use inheritance.
- Move the duplicated code into a new class, using a static operation.

## Question 10

Which of the following pattern hides the complexities of the system and provides a single point of access to the client using which the client can access the system?

### Correct Answer!

Your answer was:

- Strategy
- Factory
- Composite
- Facade
- Visitor

## Question 11

The pseudo code below is badly designed because it has not considered a pattern which?

```
class Shape {
    protected m_type;
    int m_xPos, m_yPos;

    public void Draw() {
        switch (m_type) {
            case 1:
                DrawCircle((Circle)this);
            break;
            case 2:
                DrawRect((Rect)this);
            break;
            case 3:
                DrawTriangle((Triangle)this);
            break;
        }
    }

    private void DrawCircle(Circle a_circle) {
        ...
    }

    private void DrawRect(Rect a_rect) {
        ...
    }

    private void DrawTriangle(Triangle a_rect) {
        ...
    }
}

class Circle inherits Shape {
    private float m_radius;
    ...
}

class Rect inherits Shape {
    private float m_width, m_height;
    ...
}

class Triangle inherits Shape {
    private float m_base, m_height;
    ...
}
```

Wrong Answer!

Your answer was:

- Facade
- Visitor
- Information Expert
- Polymorphism
- Strategy
- Observer

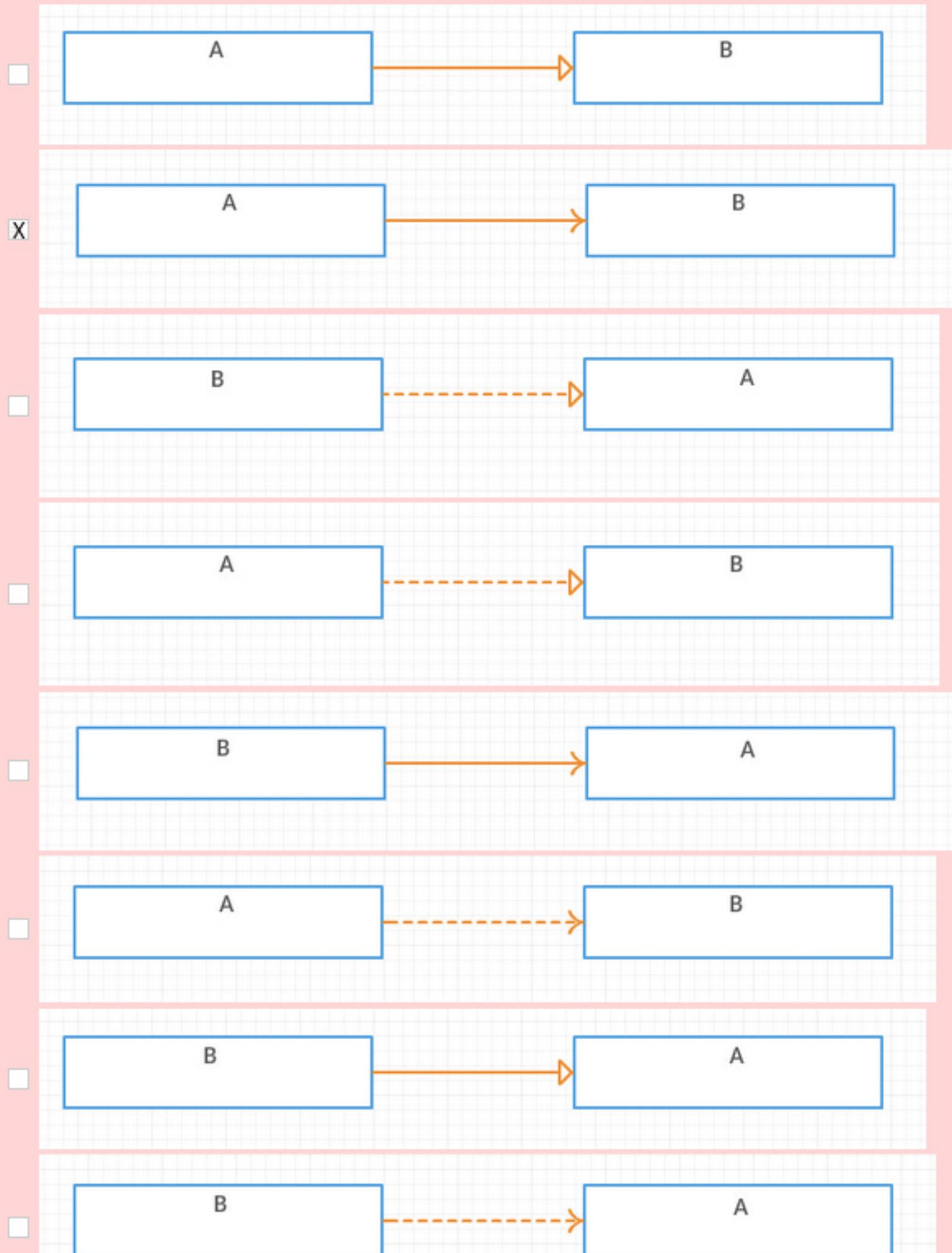
## Question 12

Given the following pseudo code for a class definition how should the relation between A and B be modeled in UML?

```
class B {  
    public void SomeOperation() {  
        ...  
    }  
}  
  
class A {  
    public B m_b;  
  
    public void DoSomething() {  
        ...  
    }  
  
    public void DoSomethingElse() {  
        ...  
        m_b.SomeOperation(this);  
        ...  
    }  
}
```

### Wrong Answer!

Your answer was:



## Question 1

Given the following pseudo code for a class definition what relation will A have to B?

```
class B {  
    public void SomeOperation(A object) {  
        object.DoSomething();  
    }  
}  
  
class A {  
    public void DoSomething() {  
        ...  
    }  
  
    public void DoSomethingElse() {  
        ...  
        B b = new B();  
        b.SomeOperation(this);  
        ...  
    }  
}
```

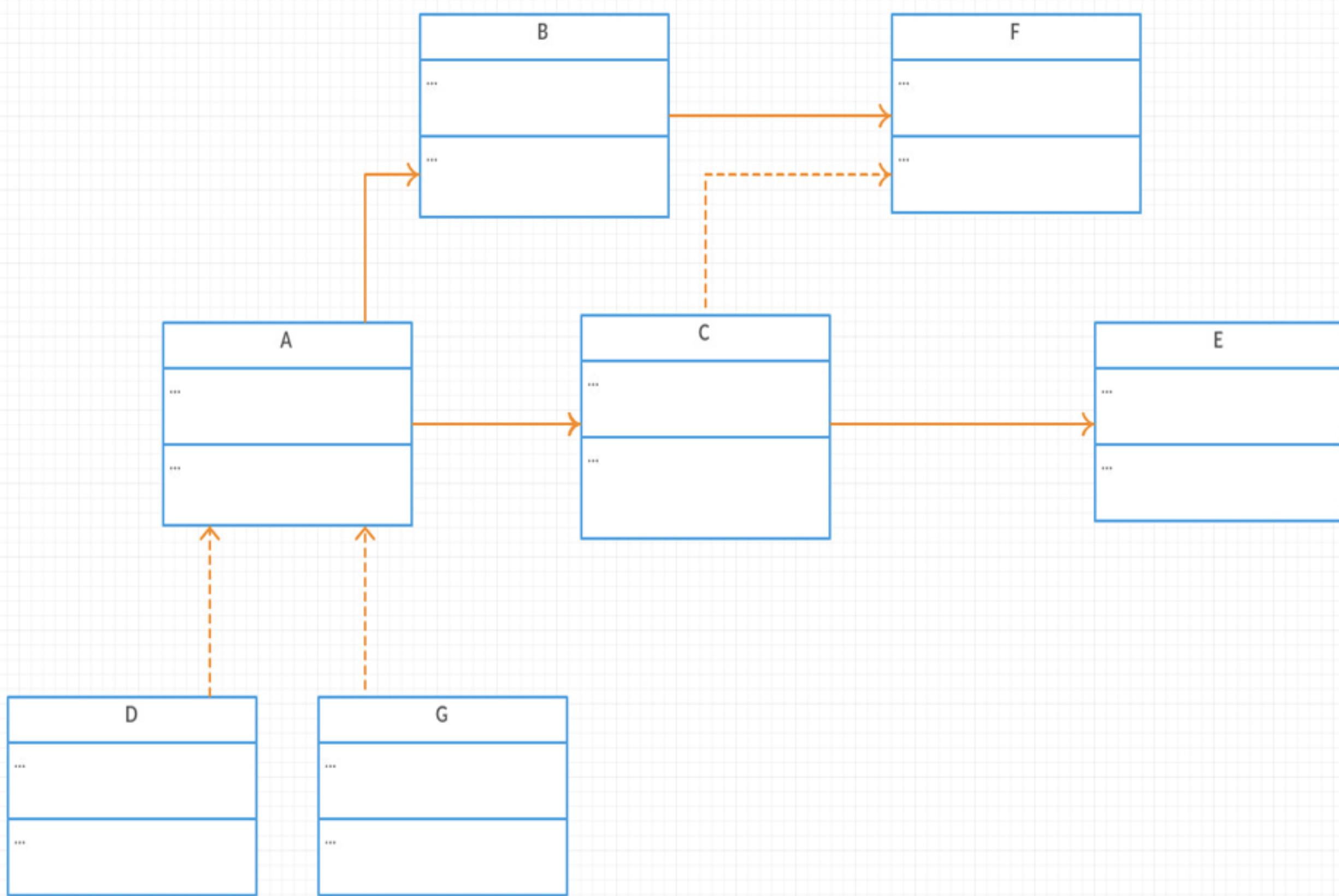
Wrong Answer!

Your answer was:

- Generalization/Specialization
- Realization
- Association
- Dependency

## Question 2

The class diagram below shows a design that uses a pattern. Which pattern is this?



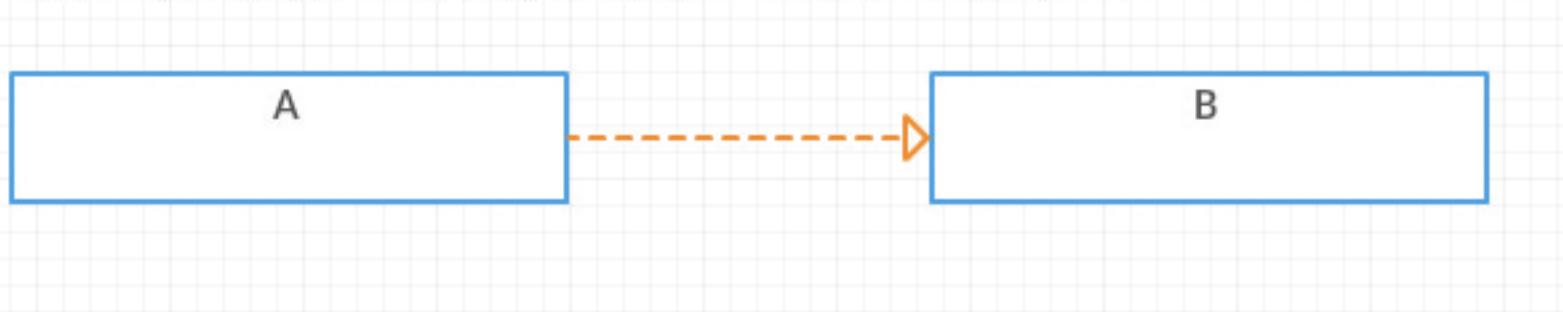
**Wrong Answer!**

Your answer was:

- Observer
- Facade
- Composite
- Visitor
- Strategy

## Question 4

The Following class diagram displays a relation between A and B, which?



**Correct Answer!**

Your answer was:

- Realization
- Generalization/Specialization
- Association
- Dependency

## Question 5

Using Composite we can

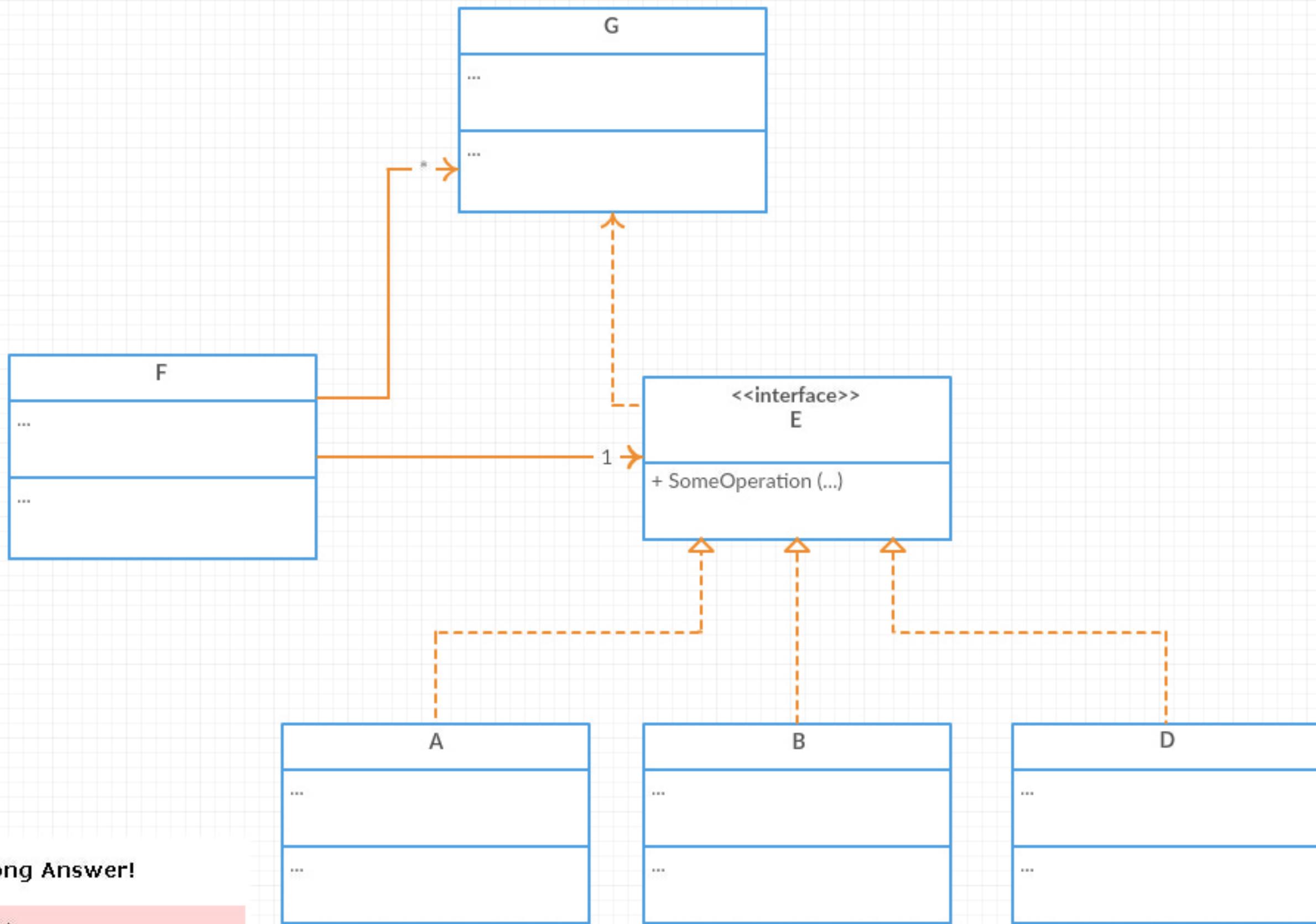
### Wrong Answer!

Your answer was:

- Provide a single point of contact to a sub-system
- Reuse algorithm variations more easily
- Treat a collection of objects as one single object
- Favor composition over inheritance
- Build recursive object hierarchies
- Create Objects

## Question 6

The class diagram below shows a design that uses a pattern. Which pattern is this?



**Wrong Answer!**

Your answer was:

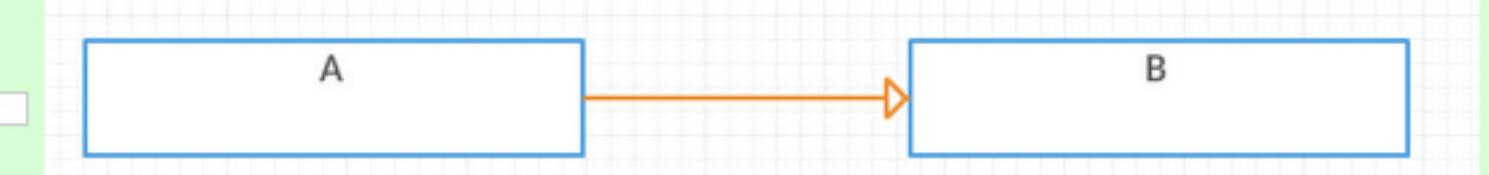
- Strategy
- Facade
- Visitor
- Composite
- Observer

## Question 7

Given the following pseudo code for a class definition how should the relation between A and B be modeled in UML?

### Correct Answer!

Your answer was:



## Question 8

Given the following pseudo code for a class definition what relation will A have to B?

```
class C {  
    public B Get() {  
        return new B();  
    }  
}  
  
class B {  
    public void SomeOperation(A object) {  
        object.DoSomething();  
    }  
}  
  
class A {  
    private C m_c;  
  
    public void DoSomething() {  
        ...  
    }  
  
    public void DoSomethingElse(B b) {  
        ...  
        m_c.Get().SomeOperation(this);  
        ...  
    }  
}
```

**Correct Answer!**

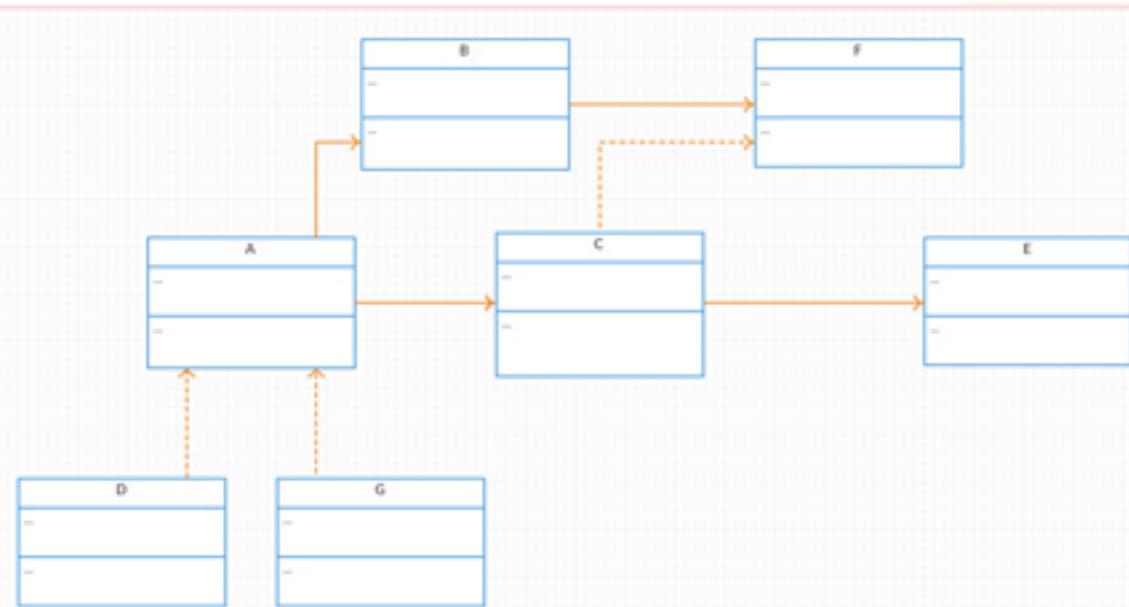
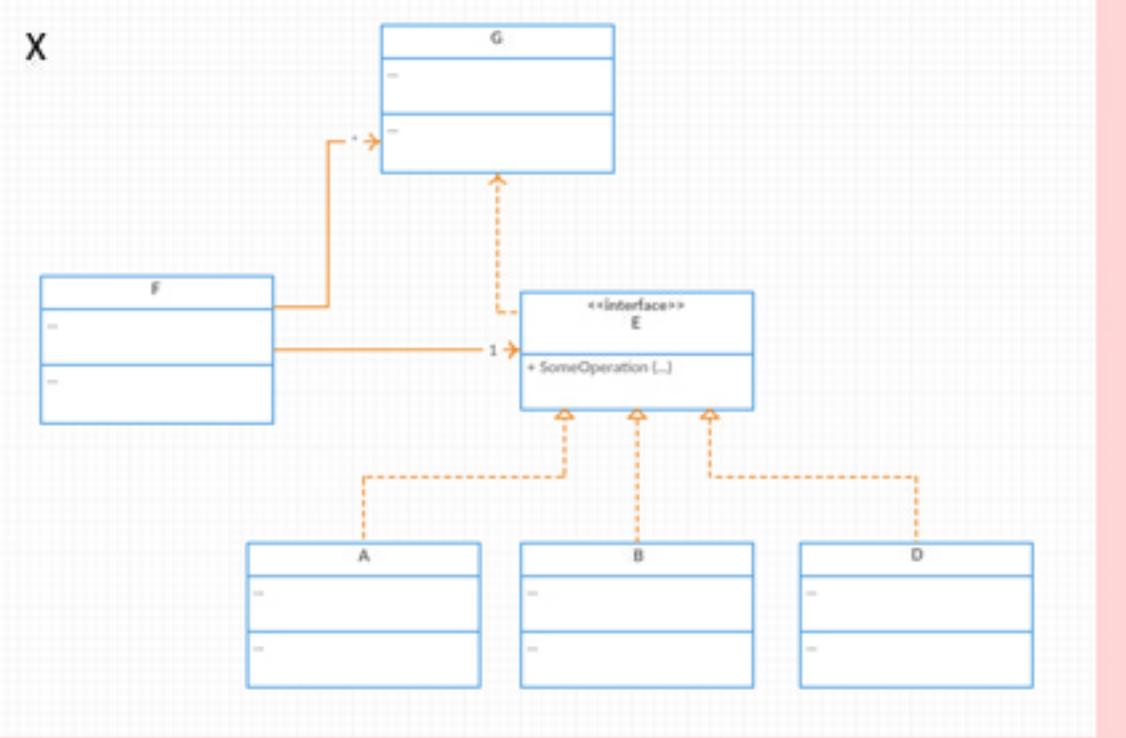
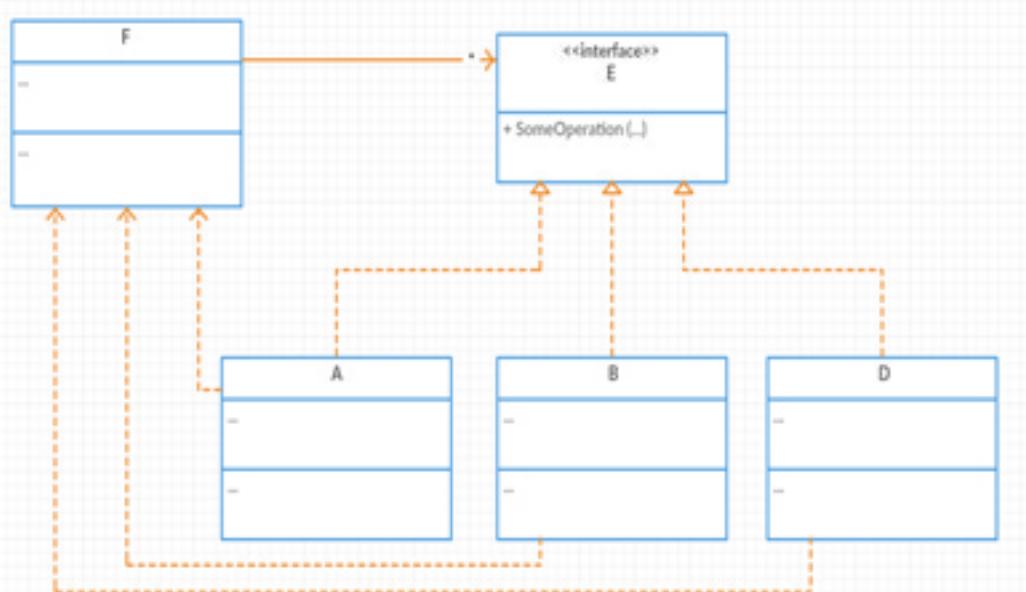
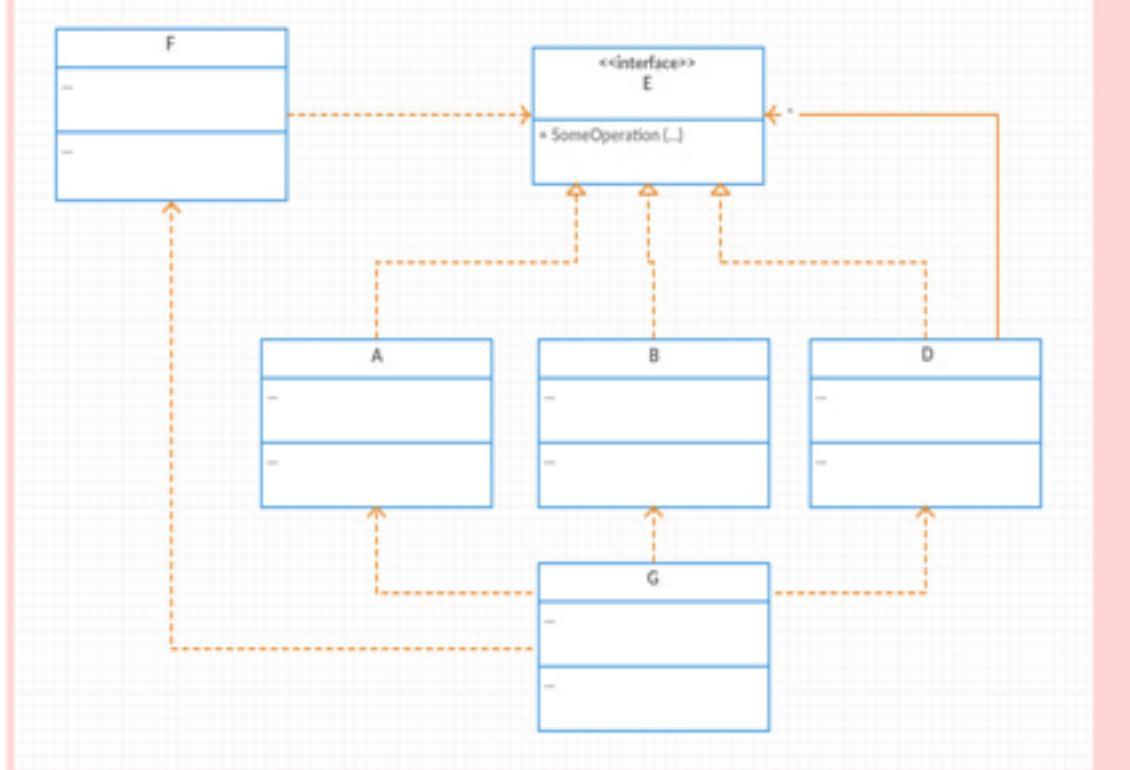
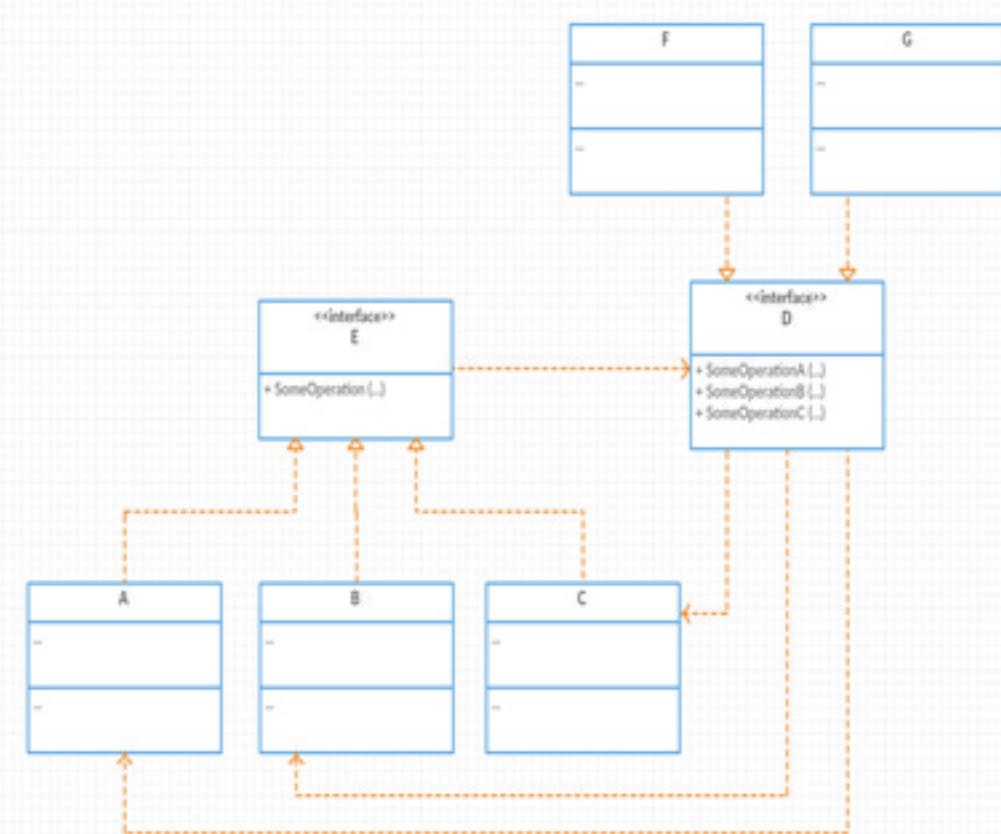
Your answer was:

- Generalization/Specialization
- Dependency
- Realization
- Association

# Question 9

Which class diagram shows the Strategy pattern.

**Wrong Answer!**



## Question 10

Given the following pseudo code for a class definition how should the relation between A and B be modeled in UML?

**Correct Answer!**

```
class B {  
    public void SomeOperation() {  
        ...  
    }  
  
class A {  
    public void DoSomething() {  
        ...  
  
    public void DoSomethingElse() {  
        ...  
        B b = new B();  
        b.SomeOperation(this);  
        ...  
    }  
}
```

Your answer was:



## Question 12

Given the following pseudo code for a class definition how should the relation between A and B be modeled in UML?

**Wrong Answer!**

```
class B {  
    public void SomeOperation() {  
        ...  
    }  
}  
  
class A {  
    public void DoSomething() {  
        ...  
    }  
  
    public void DoSomethingElse(B b) {  
        ...  
        b.SomeOperation(this);  
        ...  
    }  
}
```

Your answer was:

