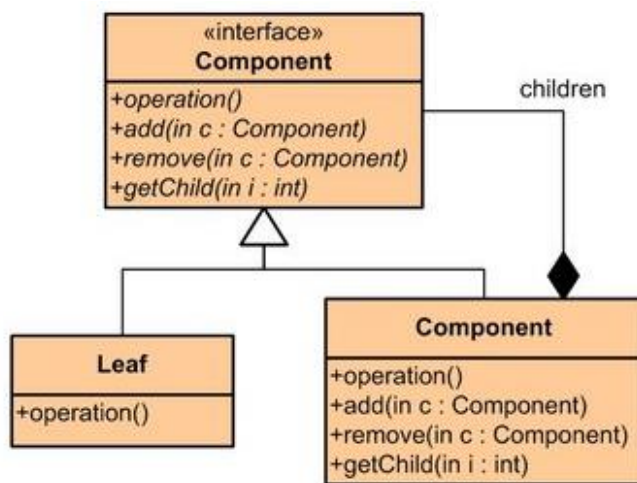


DESIGN PATTERNS: COMPOSITE

Structural Patterns

1. Design Pattern Description

Partitioning design pattern: compose objects into tree structure to represent part-whole hierarchies
→ treat compositions of objects as single instance.

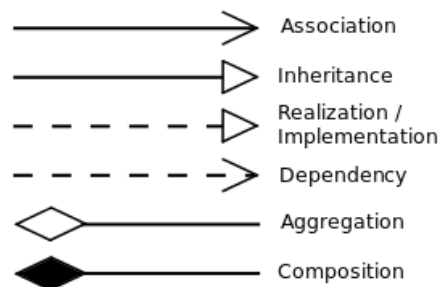


Composite

Type: Structural

What it is:

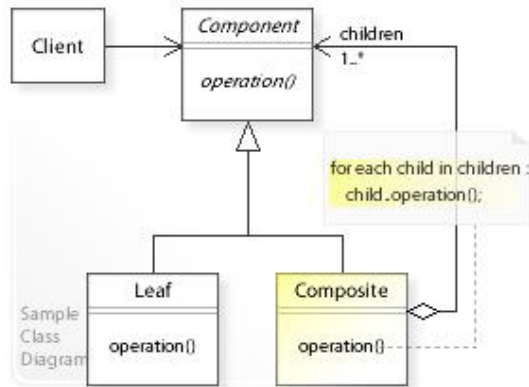
Compose objects into tree structures to represent part-whole hierarchies. Lets clients treat individual objects and compositions of objects uniformly.



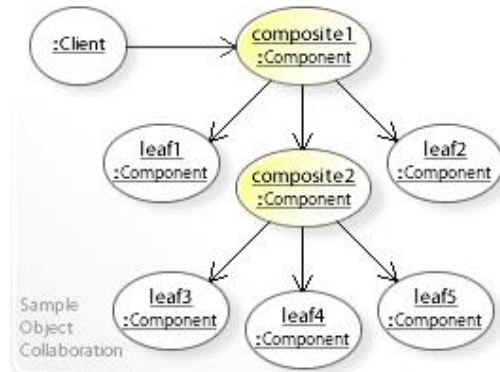
When to use?

When using multiple objects in the same way (needed nearly identical code to handle them) and clients are wanted to ignore the difference between compositions of objects and individual objects.

Class Diagram



Object Collaboration



2. Design Pattern Example

```
"""
Composite pattern example.
"""
from abc import ABC, abstractmethod

NOT_IMPLEMENTED = "You should implement this."

class Graphic(ABC):
    @abstractmethod
    def print(self):
        raise NotImplementedError(NOT_IMPLEMENTED)

class CompositeGraphic(Graphic):
    def __init__(self):
        self.graphics = []

    def print(self):
        for graphic in self.graphics:
            graphic.print()

    def add(self, graphic):
        self.graphics.append(graphic)

    def remove(self, graphic):
        self.graphics.remove(graphic)
```

```

class Ellipse(Graphic):
    def __init__(self, name):
        self.name = name

    def print(self):
        print("Ellipse:", self.name)

ellipse1 = Ellipse("1")
ellipse2 = Ellipse("2")
ellipse3 = Ellipse("3")
ellipse4 = Ellipse("4")

graphic = CompositeGraphic()
graphic1 = CompositeGraphic()
graphic2 = CompositeGraphic()

graphic1.add(ellipse1)
graphic1.add(ellipse2)
graphic1.add(ellipse3)

graphic2.add(ellipse4)

graphic.add(graphic1)
graphic.add(graphic2)

graphic.print()

```

3. Existing Example in FEM-MAT-OO

Not exactly, closest similitude in **Cost** and **Constraints** classes (actually in **CC** superclass); but they contain **ShapeFunctions** which are **NOT CHILDREN** of **CC**.

4. Design Proposal in FEM-MAT-OO

Composite design pattern can be used in **Mesh_Unfitted** superclass, since an instance of it can be an individual mesh or a composition of multiple meshes (when the resulting mesh embraces the domain limits, this can be understood as a composition of internal and boundary mesh/es).

