



# VISITOR

patron de conception comportemental

---

# INTENTIONS

---

- Sépare la structure algorithmique de la structure de données.
- Représenter une opération à effectuer sur les éléments d'une structure.
- Permet de définir une nouvelle opération sans changer le code des classes.



---

# MOTIVATIONS

---

- Quand une structure contient beaucoup de classes aux interfaces différentes
- Pour éviter la pollution des classes de la structure.
- En pratique, un visiteur permet d'obtenir le même effet que d'ajouter une nouvelle méthode virtuelle à un ensemble de classes qui ne le permet pas.

---

# AVANTAGES

---

- Ajout de nouvelles opérations très facile.
- Accumulation d'état dans le visiteur plutôt que dans des arguments.
- Permet de composer facilement des test unitaires.

---

# DÉSAVANTAGES

---

- Ajout de nouvel élément complexe.
- Le visiteur doit « casser » l'encapsulation afin de connaître la structure de l'interface.

-



---

# LIENS AVEC D'AUTRES PATRONS

---

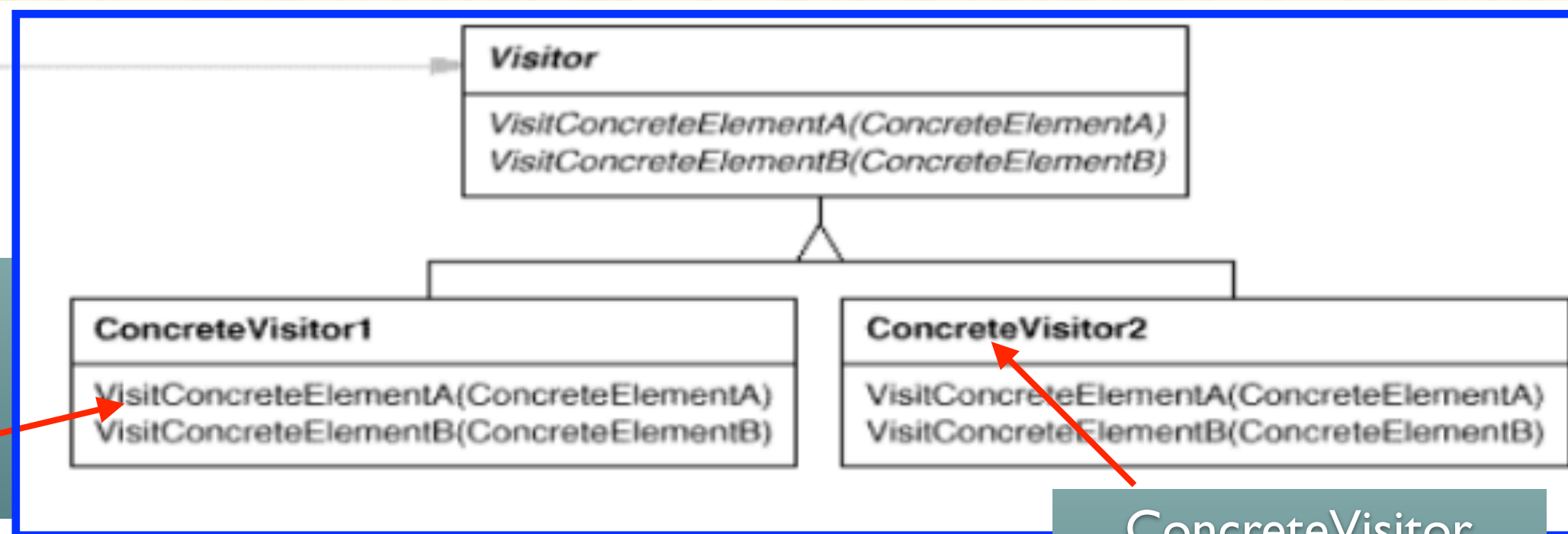
- Complète le patron d'objets composites.



# DIAGRAMME UML

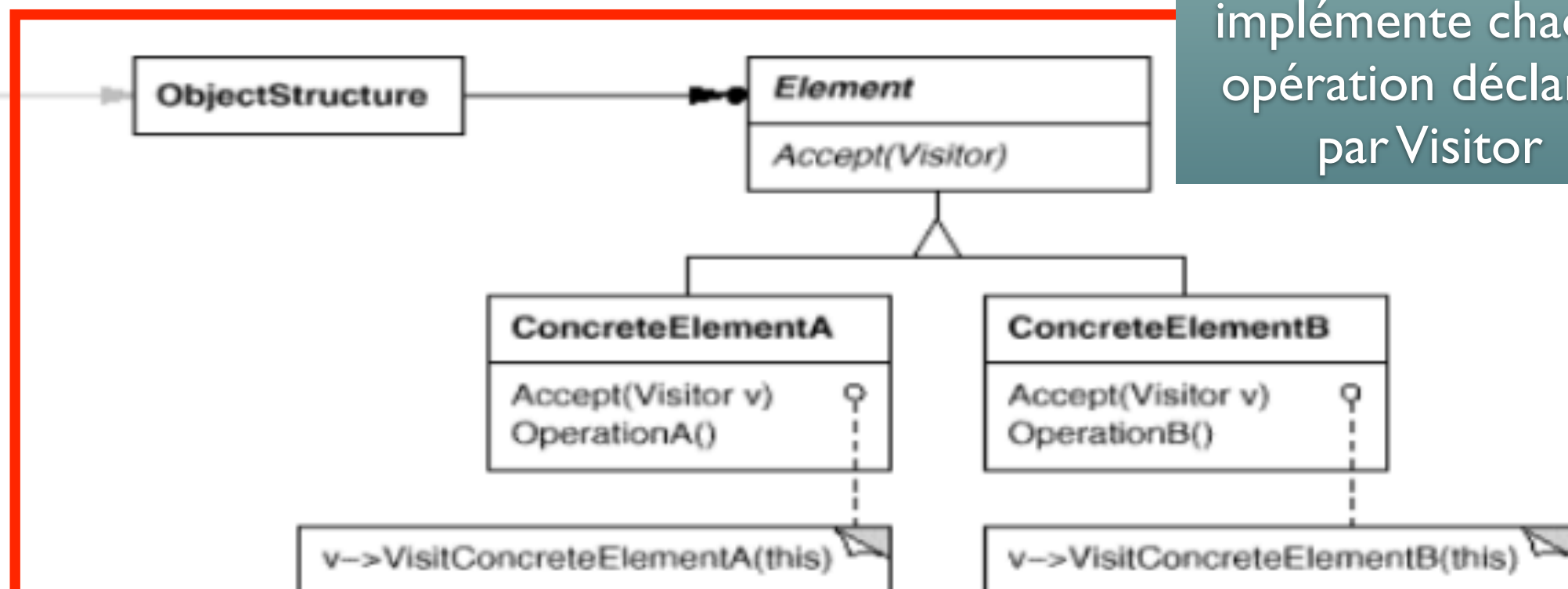
Algorithme

Chaque opération implémente un fragment de l'algorithme.



ConcreteVisitor implémente chaque opération déclarée par Visitor

Structure des données



# EXEMPLE DE CODE

```
interface MonVisiteur
{
    void visit(ObjetPere objet);
}
```

```
interface ObjetPere
{
    void accept(MonVisiteur v);
}
```

```
public class Objet1
implements ObjetPere {
    public void accept(MonVisiteur v)
    {
        v.visit(this);
    }
}
```

```
public class Objet2
implements ObjetPere
{
    public void accept(MonVisiteur v)
    {
        v.visit(this);
    }
}
```

```
public class MonVisiteurImpl1
implements MonVisiteur
{
    void visit(ObjetPere obj)
    {
        // implémentation
    }
}
```

NB :La méthode accept() doit être public



---

# EXEMPLE DE CODE

---

```
    Public static void main(String[] args)
    {
        ObjetPere objet1 = new Objet1();
        MonVisiteur marcher = new MonVisiteurImpl1();
        objet1.accept(marcher);
    }
```

---