Pattern Composite

Plan

- Définition Design Pattern
- Définition et structure du pattern Composite
- Utilisation

Qui ?

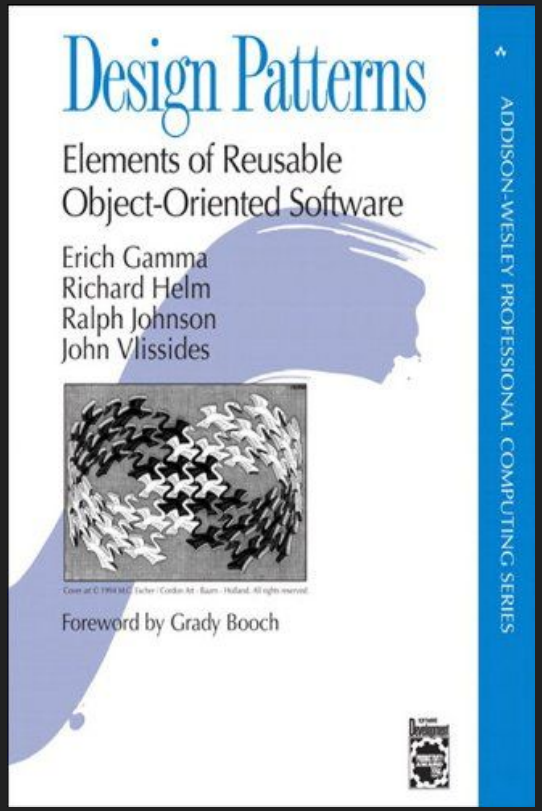
Erich Gamma

Richard Helm

Ralph E. Johnson

John Vlissides

Quoi ?


Design Patterns
Elements of Reusable
Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

La définition de Christopher Alexander

"Each pattern describes a problem which occurs over

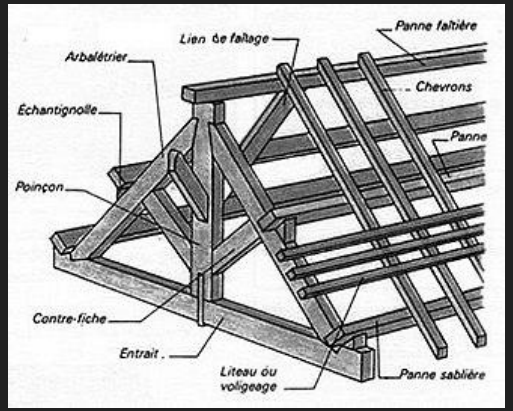
and over again in our environment, and then describes the core of the solution

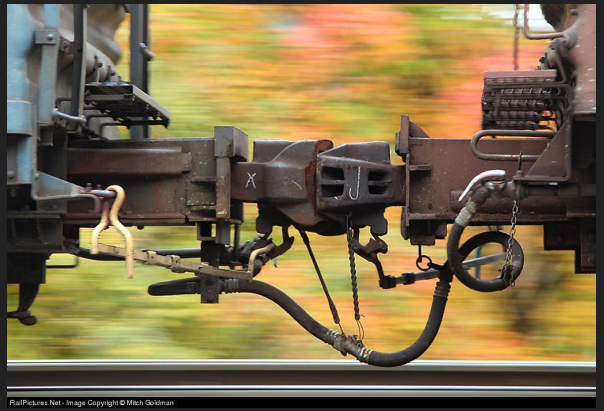
to that problem, in such a way that you can use this solution a million times

over, without ever doing it the same way twice"

Définition

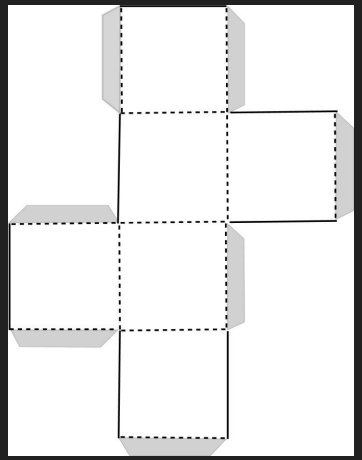
Un design pattern (ou patron de conception en français) est donc un élément ou une structure assez utilisé et réutilisé qu'il en est judicieux de le modéliser, de le théoriser et de le mettre en pratique pour adapter son utilisation répétitive.

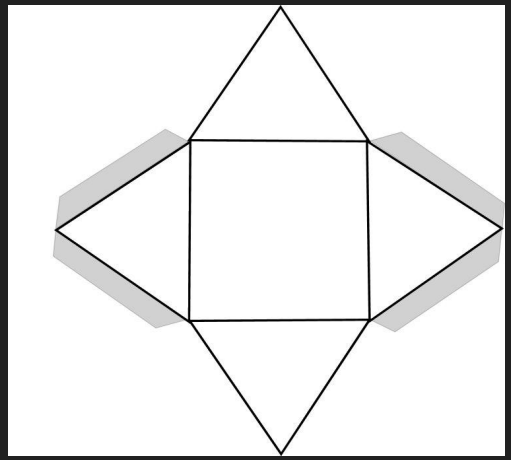




Ou encore, un ensemble d’objets attachées à des règles et contraintes simples servant à réaliser un objet entier plus complexe.

Ce qui en fait un objet réellement reconnu comme design pattern est son utilité dans des programmes différents soit sa standardité reconnue au sein du consensus des programmeurs.





. . .

Définition du Pattern Composite

-Qu’est-ce que c’est ?

-Quand l’utiliser ?

Structure du Pattern Composite

Un Component (composant)

- Abstraction de tout composants
- Fait l'interface (désigne le comportement par défaut)

Une Leaf (feuille)

- Composant sans sous-élément
- Implémente le comportement par défaut

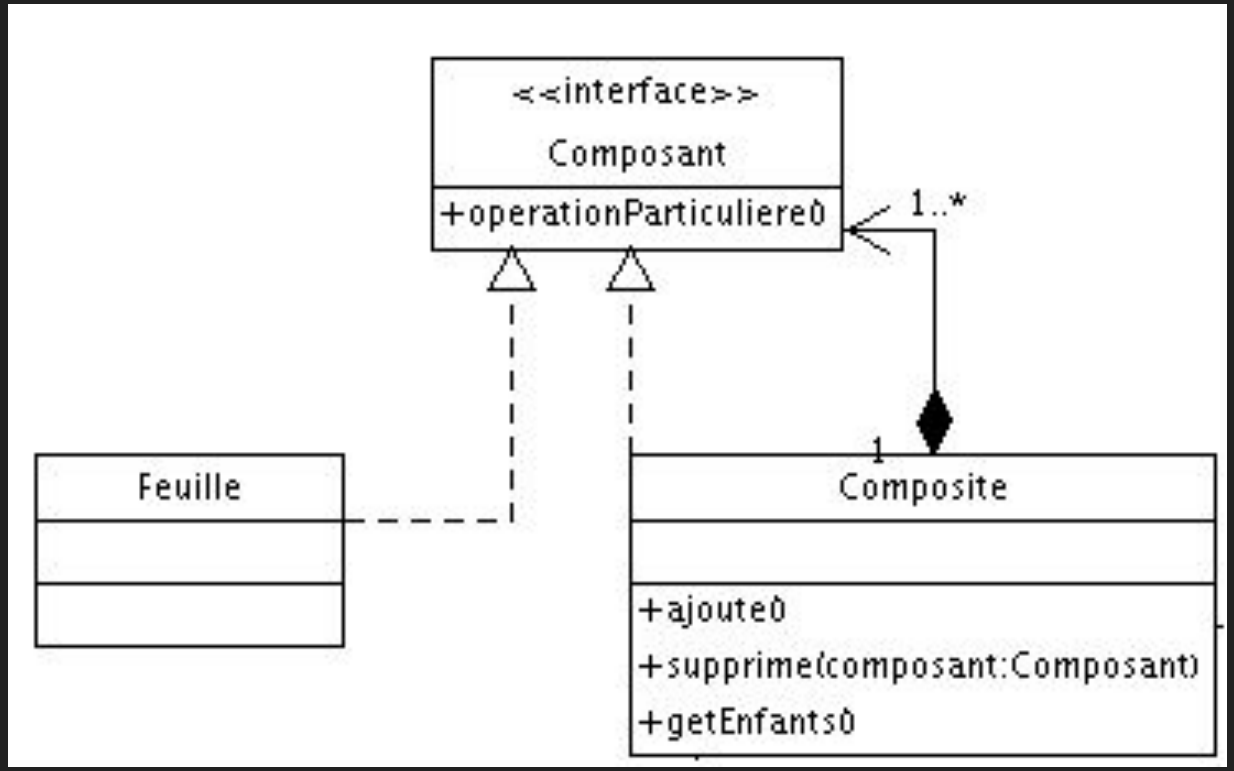
Un Composite (composé)

- Composant pouvant avoir des sous-éléments
- A des composant enfant auxquels il a accès
- Implémente un comportement à partir des enfants

Un Client

- Manipule les objets de la composition par l'interface Composant

Modélisation Théorique



Utilisation

Exemple

CHAUSSURRRE

120.00€

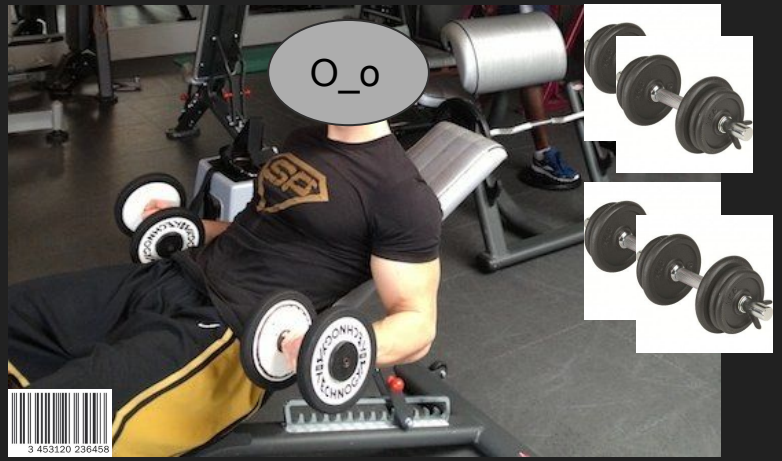
PANTALON ADID'HESS

62.00€

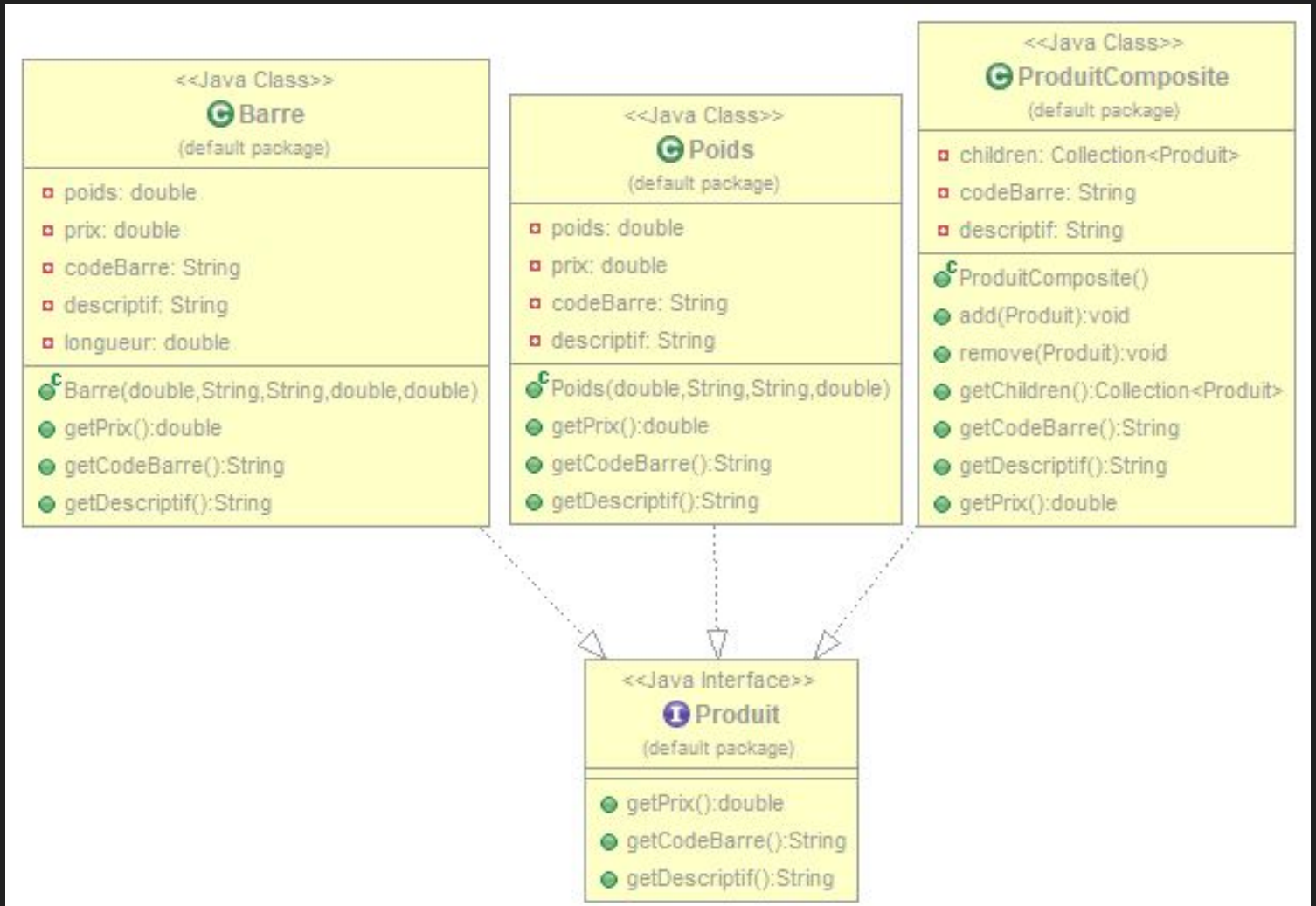
Altère Manchot

35.00€

KIT MUSCU



182.99€


<<Java Class>>
Barre
(default package)
poids: double
prix: double
codeBarre: String
descriptif: String
longueur: double
Barre(double,String,String,double,double)
getPrix():double
getCodeBarre():String
getDescriptif():String
<<Java Class>>
Poids
(default package)
poids: double
prix: double
codeBarre: String
descriptif: String
Poids(double,String,String,double)
getPrix():double
getCodeBarre():String
getDescriptif():String
<<Java Class>>
ProduitComposite
(default package)
children: Collection<Produit>
codeBarre: String
descriptif: String
ProduitComposite()
add(Produit):void
remove(Produit):void
getChildren():Collection<Produit>
getCodeBarre():String
getDescriptif():String
getPrix():double
<<Java Interface>>
Produit
(default package)
getPrix():double
getCodeBarre():String
getDescriptif():String

L’interface produit

```
public interface Produit {

    public double getPrix();
    public String getCodeBarre();
    public String getDescriptif();

}
```

La classe Barre

```java
public class Barre implements Produit {

    private double poids = 0;
    private double prix = 0;
    private String codeBarre;
    private String descriptif;
    private double longueur;

    public Barre(double prix, String descriptif, String codeBarre, double poids, double longueur) {
        this.prix=prix;
        this.descriptif=descriptif;
        this.codeBarre=codeBarre;
        this.poids=poids;
        this.longueur=longueur;
    }

    public double getPrix() {

    public String getCodeBarre() {

    public String getDescriptif() {

}
```

La classe Poids

```java
public class Poids implements Produit {

    private double poids;
    private double prix;
    private String codeBarre;
    private String descriptif;

    public Poids(double prix, String descriptif, String codeBarre, double poids) {
        this.prix=prix;
        this.descriptif=descriptif;
        this.codeBarre=codeBarre;
        this.poids=poids;
    }

    public double getPrix() {

    public String getCodeBarre() {

    public String getDescriptif() {


}
```

La classe ProduitComposite

```java
public class ProduitComposite implements Produit {

    private Collection<Produit> children;
    private String codeBarre;
    private String descriptif;

    public ProduitComposite() {
        children = new ArrayList<Produit>();
    }

    public void add(Produit produit){
        children.add(produit);
    }

    public void remove(Produit produit){
        children.remove(produit);
    }

    public Collection<Produit> getChildren() {
        return children;
    }

    public String getCodeBarre() {
        return codeBarre;
    }

    public String getDescriptif() {
        String result = new String();
        result="descriptif : (";

        for (Produit child : children) {
            result+=child.getDescriptif()+", ";
        }

        result+=")";
        return result;
    }

    public double getPrix() {
        double result = 0;
        for (Produit child : children) {
            result += child.getPrix();
        }
        result = result * 0.9;
        return result;
    }
}
```

Le main

```
public class main {

    public static void main(String[] args) {
        //                          Prix    Descriptif                  CodeBarre   Poids   Longueur
        Barre maBarre = new Barre(  25.0,   "Barre d'haltérophilie",    "BA0001",   2.0,    150.0);

        //                          Prix    Descriptif                  CodeBarre   Poids
        Poids leger = new Poids(    15.0,   "Poids d'haltère",          "PA0001",   0.5);
        Poids moyen = new Poids(    17.0,   "Poids d'haltère",          "PA0002",   1.0);
        Poids lourd = new Poids(    19.0,   "Poids d'haltère",          "PA0003",   1.5);

        ProduitComposite haltere = new ProduitComposite();
        haltere.add(maBarre);
        haltere.add(leger);
        haltere.add(moyen);
        haltere.add(lourd);

    }
}
```

QCM

Question 1 : Une feuille c’est :

- Un composant composé d’un seul élément

- Un composant qui n’a pas de sous élément

- Le truc qui tombe des arbres en automne

QCM

Question 1 : Une feuille c’est :

- Un composant composé d’un seul élément

- Un composant qui n’a pas de sous élément

- Le truc qui tombe des arbres en automne

QCM

Question 2 : Un composite :

-Se compose d’un ou plusieurs éléments

-Compose un élément

-Compose seulement s’il en a envie

QCM

Question 2 : Un composite :

-Se compose d’un ou plusieurs éléments

-Compose un élément

-Compose seulement s’il en a envie

QCM

Question 3 : Un client :

-Est un composant particulier

-Manipule les objets de la composition par l’interface composant

-Va acheter son lot d’haltères

QCM

Question 3 : Un client :

-Est un composant particulier

-Manipule les objets de la composition par l’interface composant

-Va acheter son lot d’haltères

QCM

Question 4 : Le nom du livre du “The Gang Of Four” est :

-“Design Patterns, Elements of Reusable Object Oriented Software”

-“Design Patterns, Reusable Elements of Object Oriented Software”

-“Design Patterns, Elements of Reusable Object Software Oriented”

QCM

Question 4 : Le nom du livre du “The Gang Of Four” est :

-“Design Patterns, Elements of Reusable Object Oriented Software”

-“Design Patterns, Reusable Elements of Object Oriented Software”

-“Design Patterns, Elements of Reusable Object Software Oriented”

Merci !