Conception et Programmation objet Avancées

Décorateur

Sylvie Coste

IUT de Lens

2021–2022

Plan

Décorateur

Un problème de glace ...

On souhaite réaliser une application qui gère la vente de glaces. Elle doit permettre d'afficher dans la console le nom complet de la glace achetée ainsi que son prix.

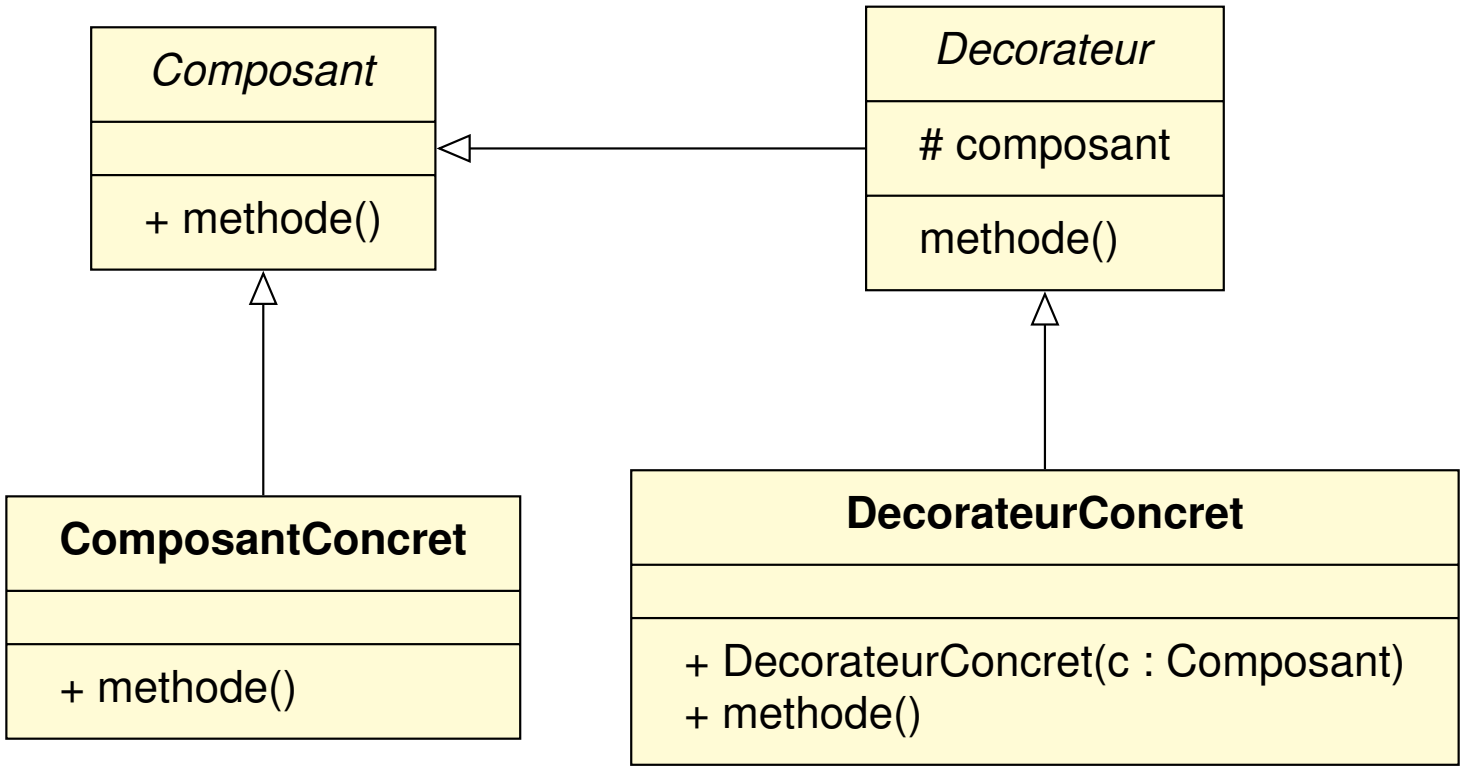
Les clients ont le choix entre des glaces à l'eau et des glaces à l'italienne.

Toutes ces glaces peuvent avoir les mêmes parfum : vanille, fraise et citron.

Des options (amandes grillées, chantilly) peuvent être ajoutées.

- Comment modéliser cela ?
- De combien de classes aurez-vous besoin ?

Diagramme de classes



Design pattern décorateur

Objectif

- Associer dynamiquement des responsabilités supplémentaires à un objet

Problème

- L'objet possède les fonctions de base. Des fonctionnalités supplémentaires s'appliqueront avant ou après

Solution

- Permettre l'extension de la fonctionnalité d'un objet sans avoir recours à des sous-classes explicites

Conséquence

- La fonctionnalité à ajouter se trouve dans des composants de petite taille, ce qui permet au décorateur de l'ajouter dynamiquement avant ou après

Les constituants du DP décorateur (1)

Composant

- Peut être utilisé seul ou enveloppé par un Décorateur

ComposantConcret

- Hérite de Composant
- Objet auquel on ajoute dynamiquement un nouveau comportement

Les constituants du DP décorateur (2)

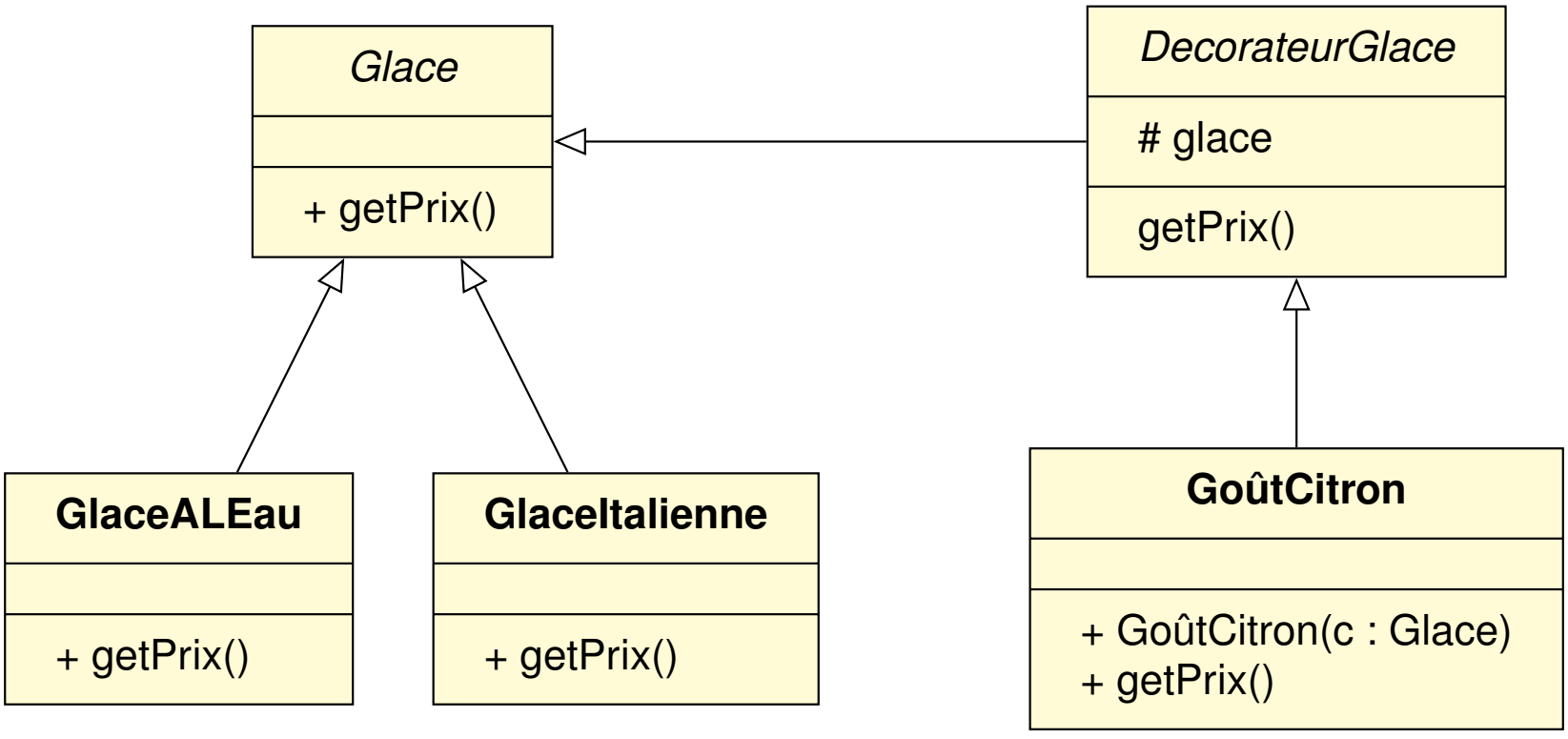
Décorateur

- Étend le composant

DécorateurConcret

- Possède une variable d'instance pour l'objet qu'il décore
- Peut étendre l'état du composant
- Peut ajouter de nouvelles méthodes

Diagramme de classes du problème de glaces (extrait)



Indications d'utilisation du DP décorateur

- Pour ajouter dynamiquement des responsabilités à des objets individuels de manière transparente (sans affecter les autres objets)
- Pour ajouter des responsabilités qu'on veut pouvoir retirer
- Quand l'extension par dérivation est impraticable (trop grand nombre d'extensions possibles, par exemple)

Décorateur : savez-vous le programmer ?

Programmer le DP décorateur

- Coder la classe abstraite Glace
- Coder la classe GlaceItalienne
- Coder la classe abstraite DecorateurGlace
- Coder la classe ParfumCitron
- Créer une glace à l'eau parfum citron et une glace italienne parfum fraise avec de la Chantilly et des amandes grillées, afficher leur prix et leur description

Le code du décorateur (1)

La classe abstraite Glace

```
public abstract class Glace {
    private String libelle;
    private double prix;
    public String getLibelle() {
        return libelle;
    }
    public double getPrix() {
        return prix;
    }
    public void setLibelle(String libelle) {
        this.libelle = libelle;
    }
    public void setPrix(double prix) {
        this.prix = prix;
    }
    public String toString() {
        return String.format("Glace %s, prix %s",
                             getLibelle(),getPrix());
    }
}
```

Le code du décorateur (2)

La classe GlaceItalienne

```
public class GlaceItalienne extends Glace {
    public GlaceItalienne() {
        setLibelle("à l'italienne");
        setPrix((3.5));
    }
}
```

Le code du decorateur (3)

La classe abstraite DecorateurGlace

```
public abstract class DecorateurGlace extends Glace{
    protected Glace glace;
    public abstract String getLibelle();
    public abstract double getPrix();
}
```

La classe GoûtCitron

```
public class GoutCitron extends DecorateurGlace {
    public GoutCitron(Glace g) {
        glace = g;
    }
    public String getLibelle() {
        return glace.getLibelle()+" goût citron";
    }
    public double getPrix() {
        return glace.getPrix()+0.5;
    }
}
```

Le code du decorateur (4)

Utilisation

```
Glace g1 = new GoutCitron(new GlaceALEau());
System.out.println(g1);

Glace g2 = new IngredientAmandes(
                new IngredientChantilly(
                new GoutFraise(new GlaceItalienne())));
System.out.println(g2);
```