DESIGN PATTERN FLYWEIGHT

(Pattern poids-mouche)

Kalvin NARCEAU Gaël DIM AMORIN Dylan ESTRADE Clément

PLAN

A. Qu'est ce que le design pattern Flyweight ?
   a. Rappel : Design Pattern
   b. Notre problématique
   c. Comment résoudre cette problématique ?

B. Etude d'un cas pratique

A. Qu'est ce que le design Pattern Flyweight ?

a. Rappel : Design Pattern

Problème Récurrent

Langage Orienté Objet

Pattern de :
- Structuration
- Construction
- Comportement

Nom

Problématique

Solution

Conséquences

A. Qu'est ce que le design Pattern Flyweight ?

b. La problématique du Pattern

BEAUCOUP de petits Objets

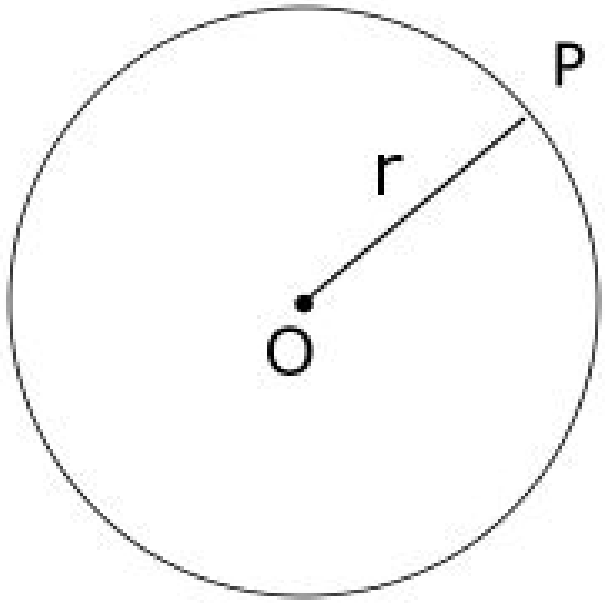
Attribut(s) redondant(s)

Coûte du temps

Coûte de la mémoire

A. Qu'est ce que le design Pattern Flyweight ?

b. La problématique du Pattern



Etat Extrinsèque : le rayon du cercle

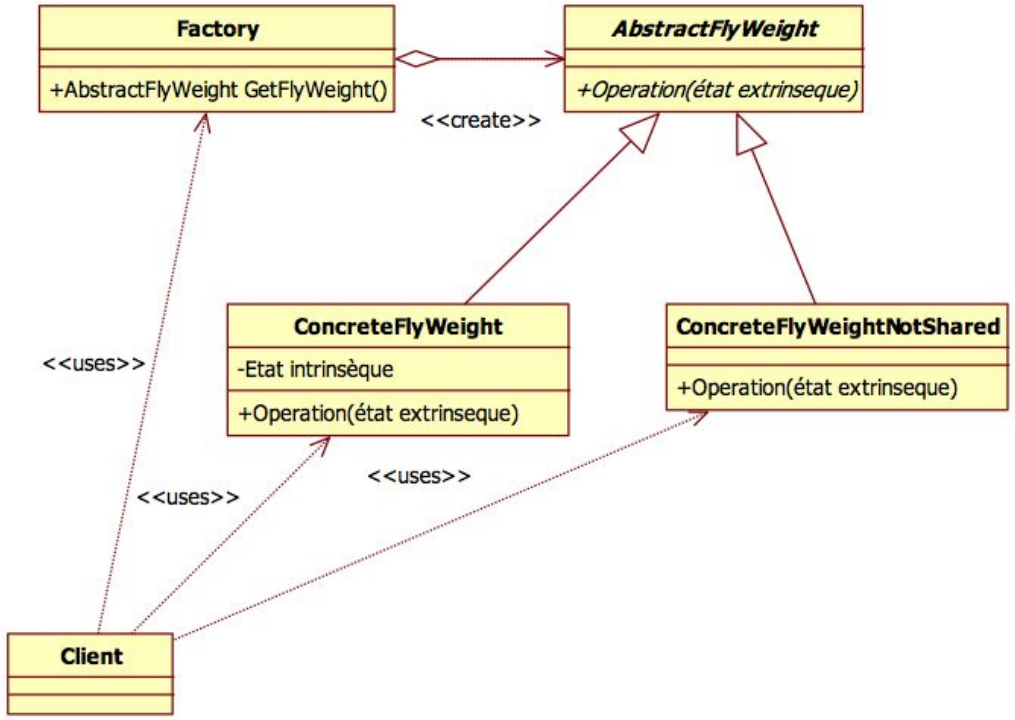
Entre 0 et l'infini

Etat Intrinsèque : la couleur du cercle

BLEU *ou* ***ROUGE***

A. Qu'est ce que le design Pattern Flyweight ?

c. Comment résoudre ce problème ?



Liste d'objets déjà créés

Demande d'un objet à Factory :

Si l'objet est dans la liste, on le renvoie sans en créer un nouveau

Sinon il est créé dans la liste.

Cas pratique

Class People

```
class People{
    private PencilFactory pencilFactory;
    public void borrow(final String[] colors){
        for(final String c : colors){
            System.out.println("Intended color : "+c);
            System.out.println("Received pencil instance : "+this.pencilFactory.borrowPencil(c));
        }
    }
    public void setPencilFactory(final PencilFactory pencilFactory){
        this.pencilFactory=pencilFactory;
    }
}
```

Kalvin NARCEAU Gaël DIM AMORIN Dylan ESTRADE Clément

Interface et class qui implémente

```java
interface WrittingObject {

}


class Pencil implements WrittingObject {
    private final String color;
    private final int id;
    public Pencil(final String color){
        this.color=color;
        this.id=new Random().nextInt();
    }
    @Override
    public String toString(){
        return "Pencil's instance for color : "+this.color+", id "+this.id;
    }
}
```

Kalvin NARCEAU Gaël DIM AMORIN Dylan ESTRADE Clément

Class Entreprise

```
class PencilFactory{
    private final Map<String, Pencil> pencils=new HashMap<String, Pencil>();
    public Pencil borrowPencil(final String color){
        if(!this.pencils.containsKey(color)){
            this.pencils.put(color, new Pencil(color));
        }
        return this.pencils.get(color);
    }
    public int getPencilsSize(){
        return this.pencils.size();
    }
}
```

Kalvin NARCEAU Gaël DIM AMORIN Dylan ESTRADE Clément

Conclusion

Kalvin NARCEAU Gaël DIM AMORIN Dylan ESTRADE Clément