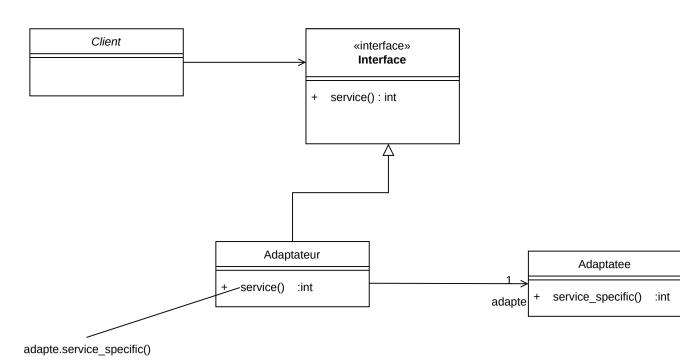
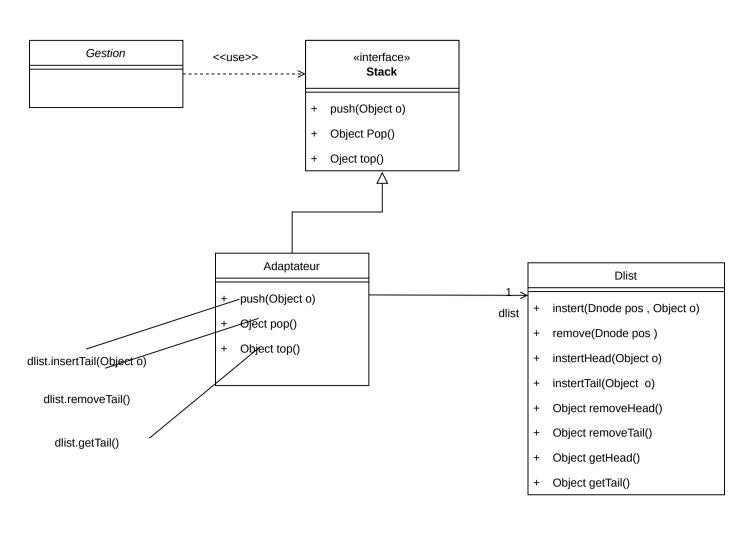
II- Pattern de structuration

1.1- Pattern Adaptateur

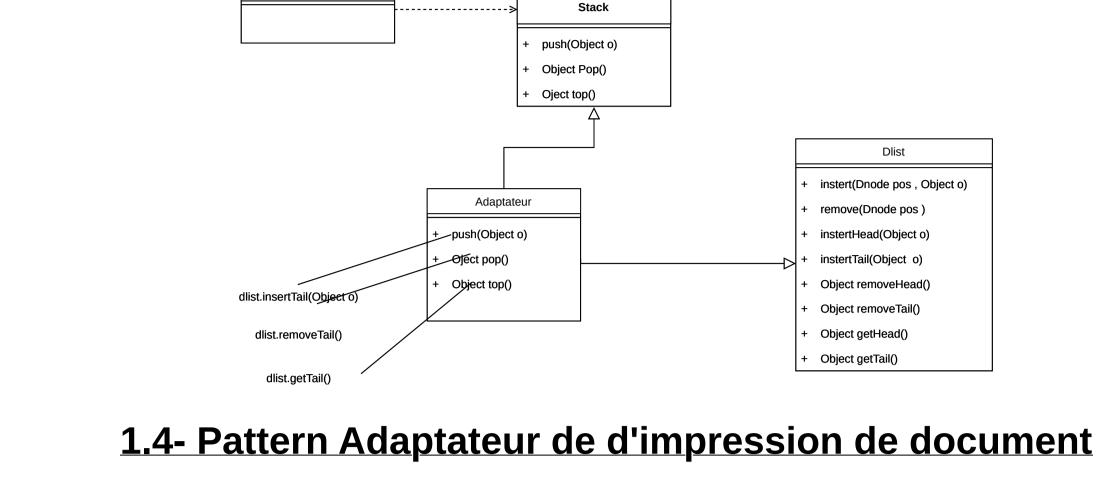


1.2- Pattern Adaptateur de Dlist avec adapteur implementant Dlist (Exemple 2)



1.3- Pattern Adaptateur de Dlist avec adapteur Heritant de Dlist

«interface»



<<use>>

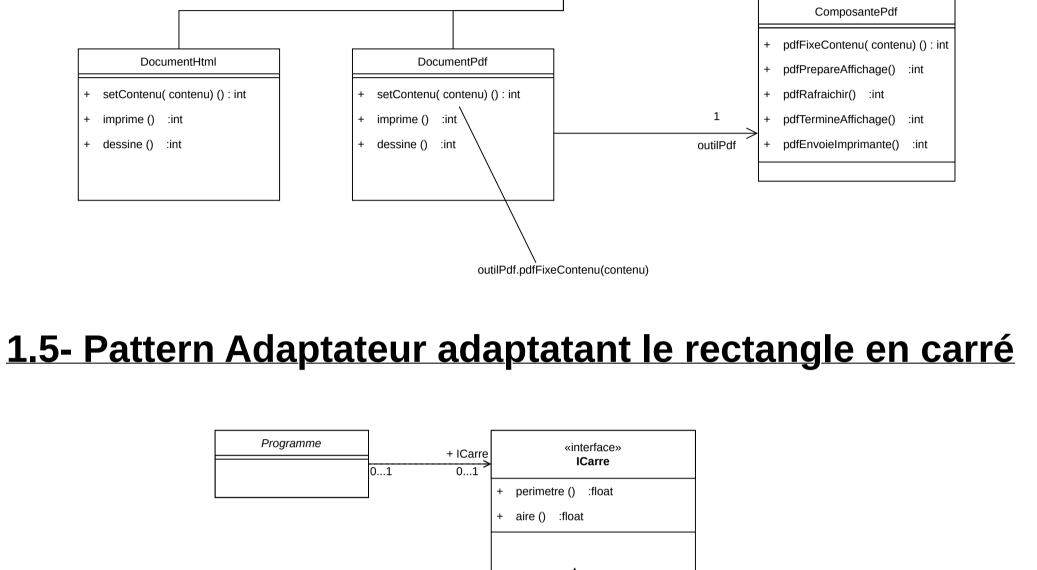
Gestion

ServeurWeb

+ imprime () :int + dessine () :int

+ setContenu(contenu) () : int

«interface» **Document**



Rectangle

- longueur: floate

largeur: floate

perimetre() :float

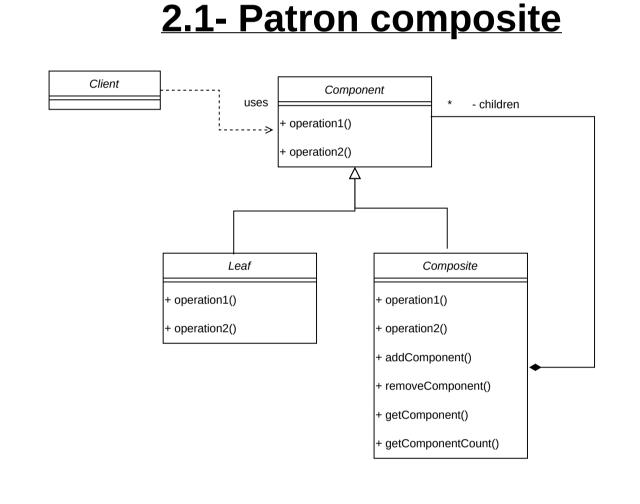
aire() :float

+ aire () :float + rectangle 0..1

AdapteurRectangle

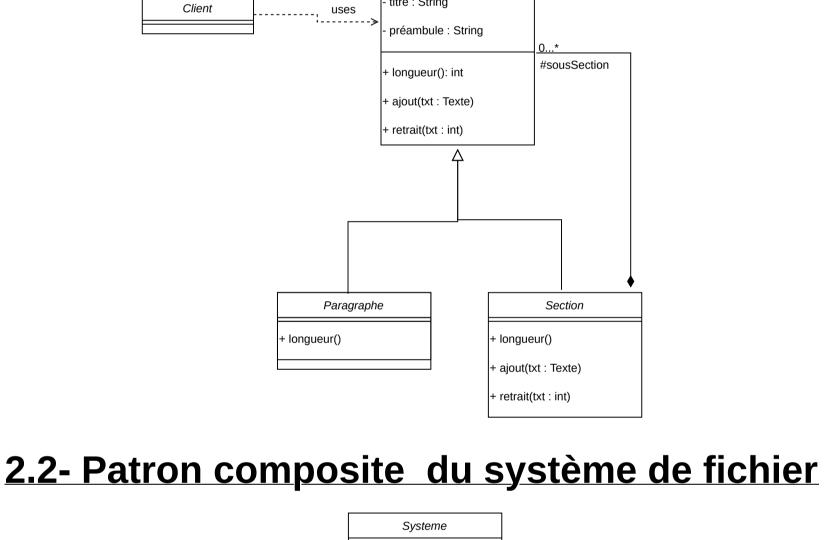
rectangle.setLong(cote); retangle.setLarg(cote); retangle.perimetre()

+ perimetre() : float



Texte titre : String uses

2.2- Patron composite Exemple avec le Texte



....uses :-----> type : String #sousSection + decrire() ajouter(elm:Systeme)

suprimer(elm:systeme)

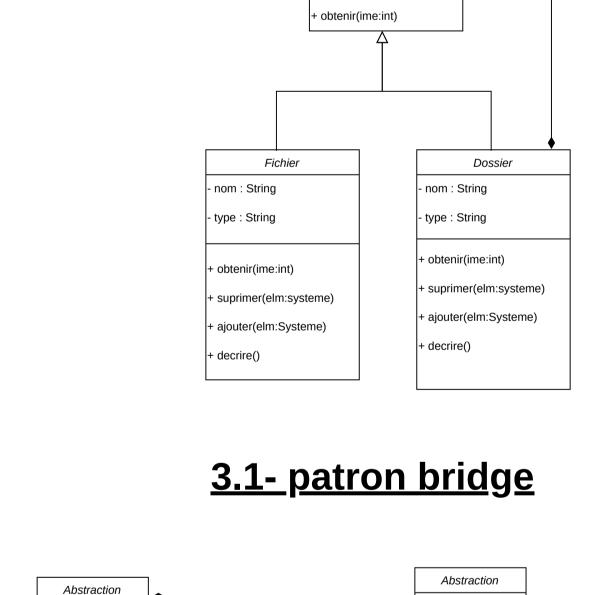
nom : String

Client

Operation()

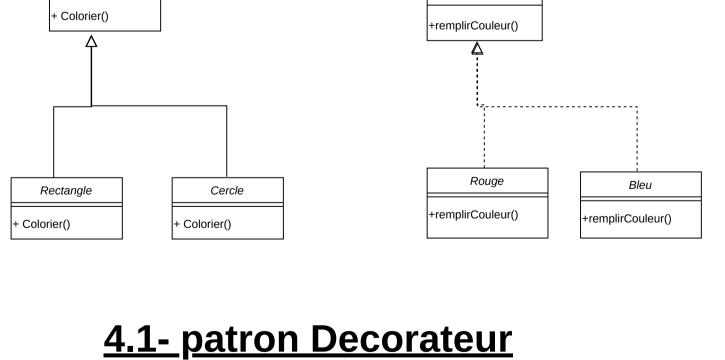
RefinedAbstraction

Operation()



<<Interface>> + couleur Formes **Abstraction**

3.2- patron bridge pour les forme



Operation()

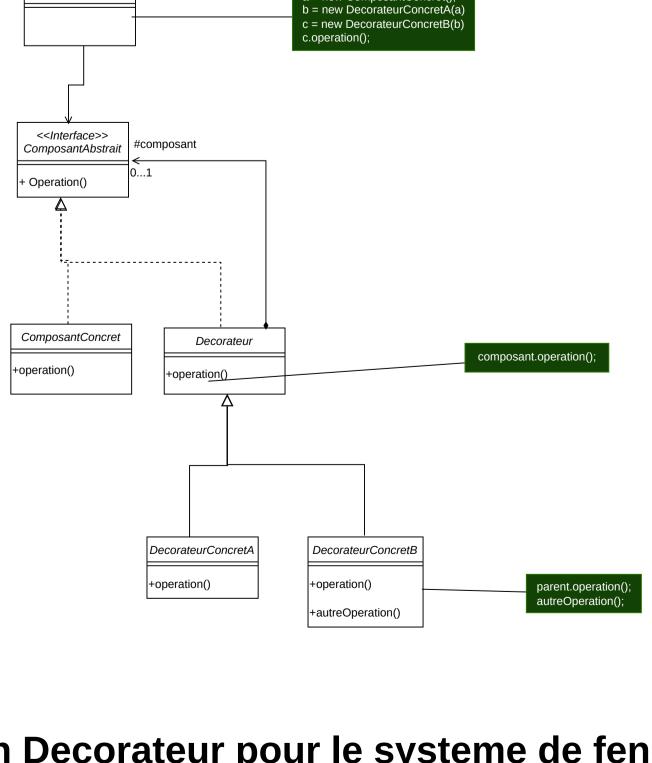
ConcreteImplementorA

OperationImp()

ConcreteImplementorB

OperationImp()

Client a = new ComposantConcret();



Client

