

AULA BRIDGE

PONTES SÃO IMPORTANTES.

NÃO QUEIME PONTES!





OBJETIVOS

- APRESENTAR O PROBLEMA GERAL
- APRESENTAR UMA SOLUÇÃO UTILIZANDO O BRIDGE
- CONHECER COMO ESTE PADRÃO PODE SER UTILIZANDO JUNTAMENTE COM UM EXEMPLO SIMPLES DE BACKEND

PROBLEMAS

- COMO É POSSÍVEL FAZER COM QUE A ABSTRAÇÃO E A IMPLEMENTAÇÃO POSSAM VARIAR INDEPENDENTEMENTE?
- COMO ESTA IMPLEMENTAÇÃO PODE VARIAR EM TEMPO DE EXECUÇÃO?



SOLUÇÃO

- DEFINIR UM CONJUNTO HIERÁRQUICO PARA AMBOS OS LADOS: ABSTRAÇÃO E IMPLEMENTAÇÃO
- EM TEMPO DE EXECUÇÃO, SERÁ POSSÍVEL ESCOLHER A CLASSE CONCRETA PARA A ABSTRAÇÃO E PARA A IMPLEMENTAÇÃO...
- ... QUE SÃO COMPATÍVEIS GRAÇAS ÀS INTERFACES

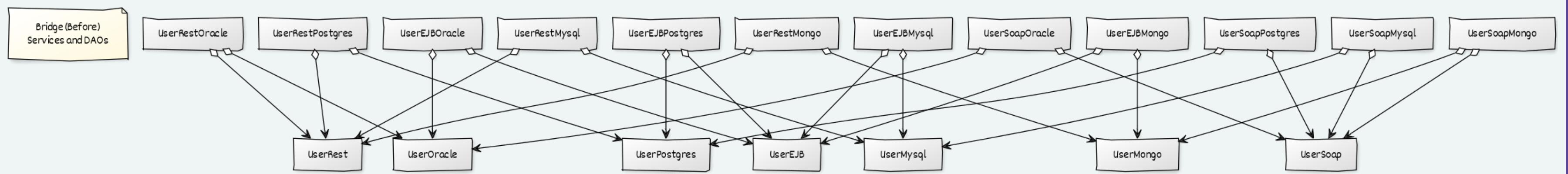
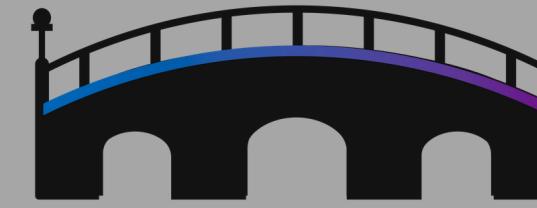


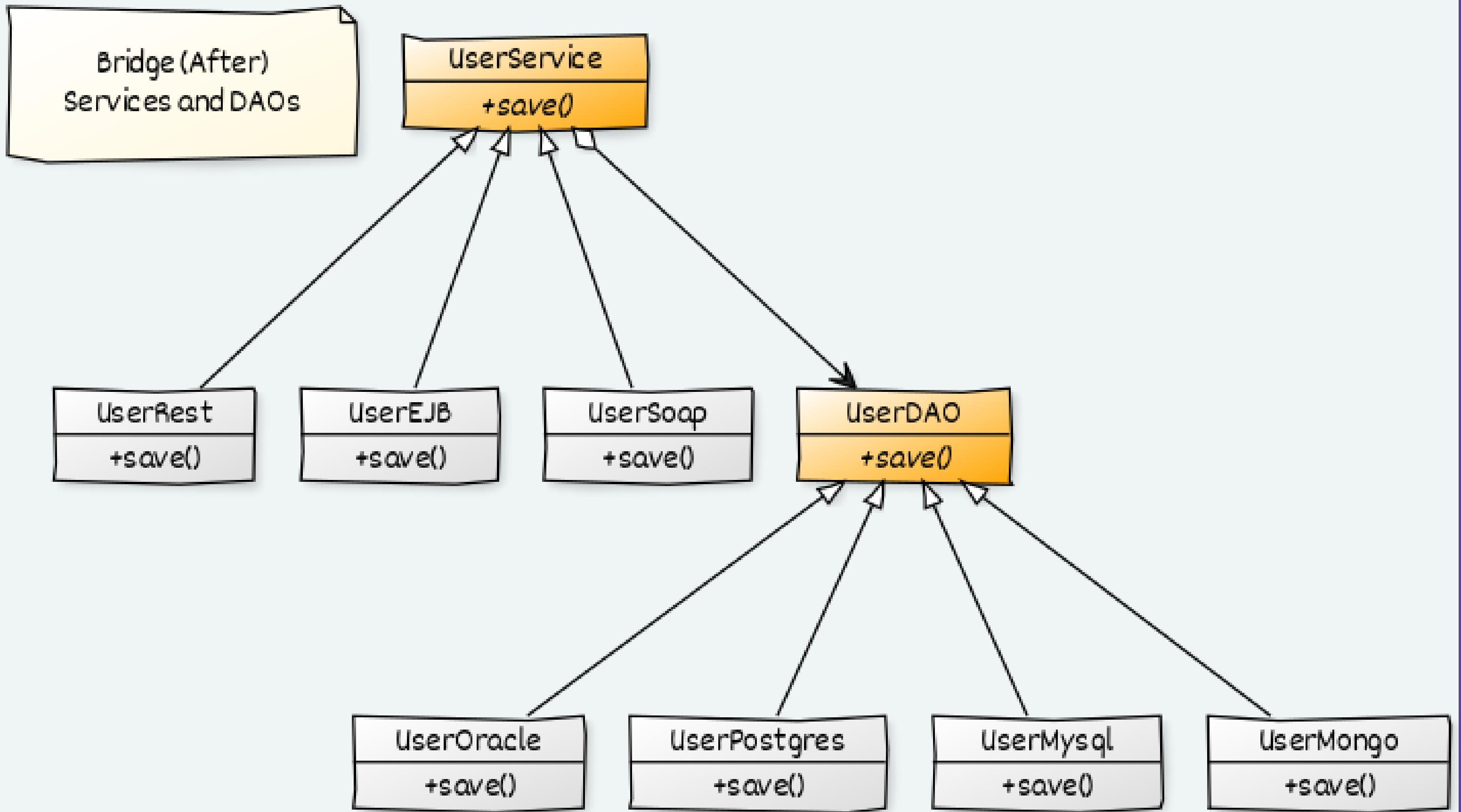


Desacoplar uma abstração de sua implementação para que os dois possam variar independentemente.

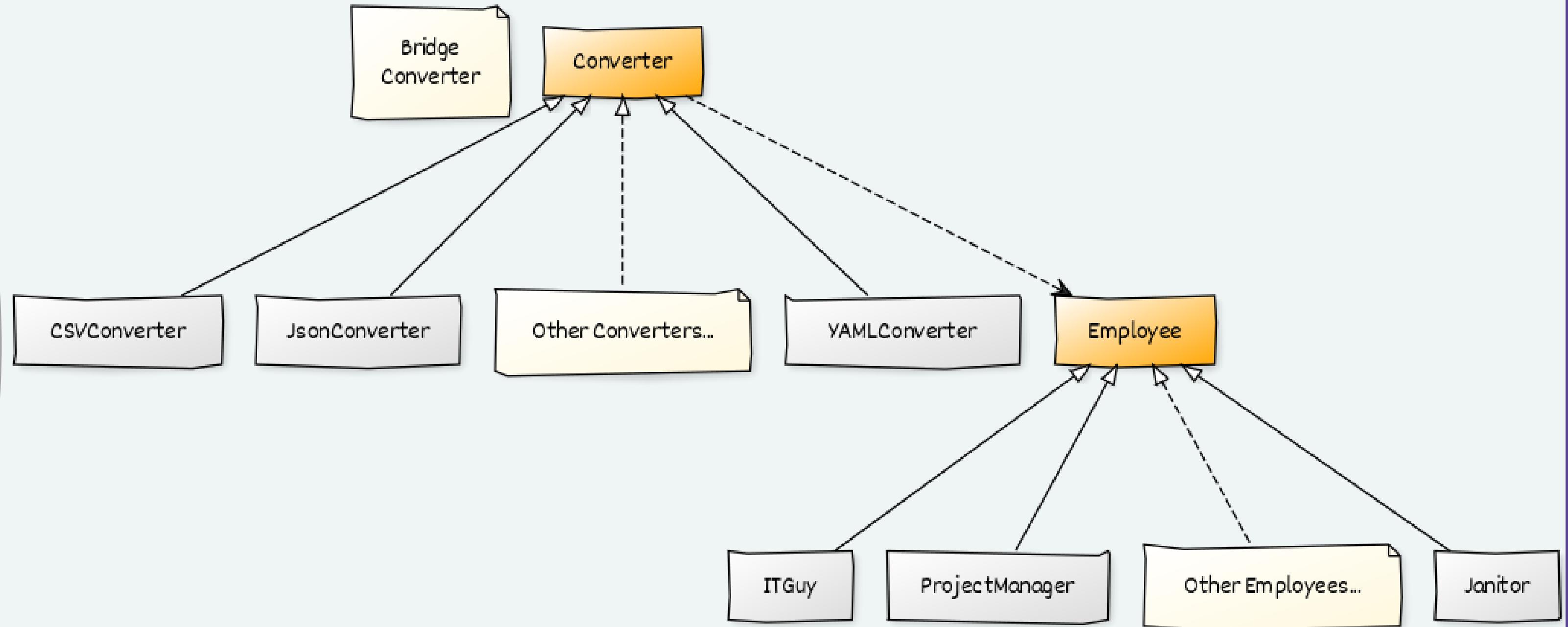
GOF







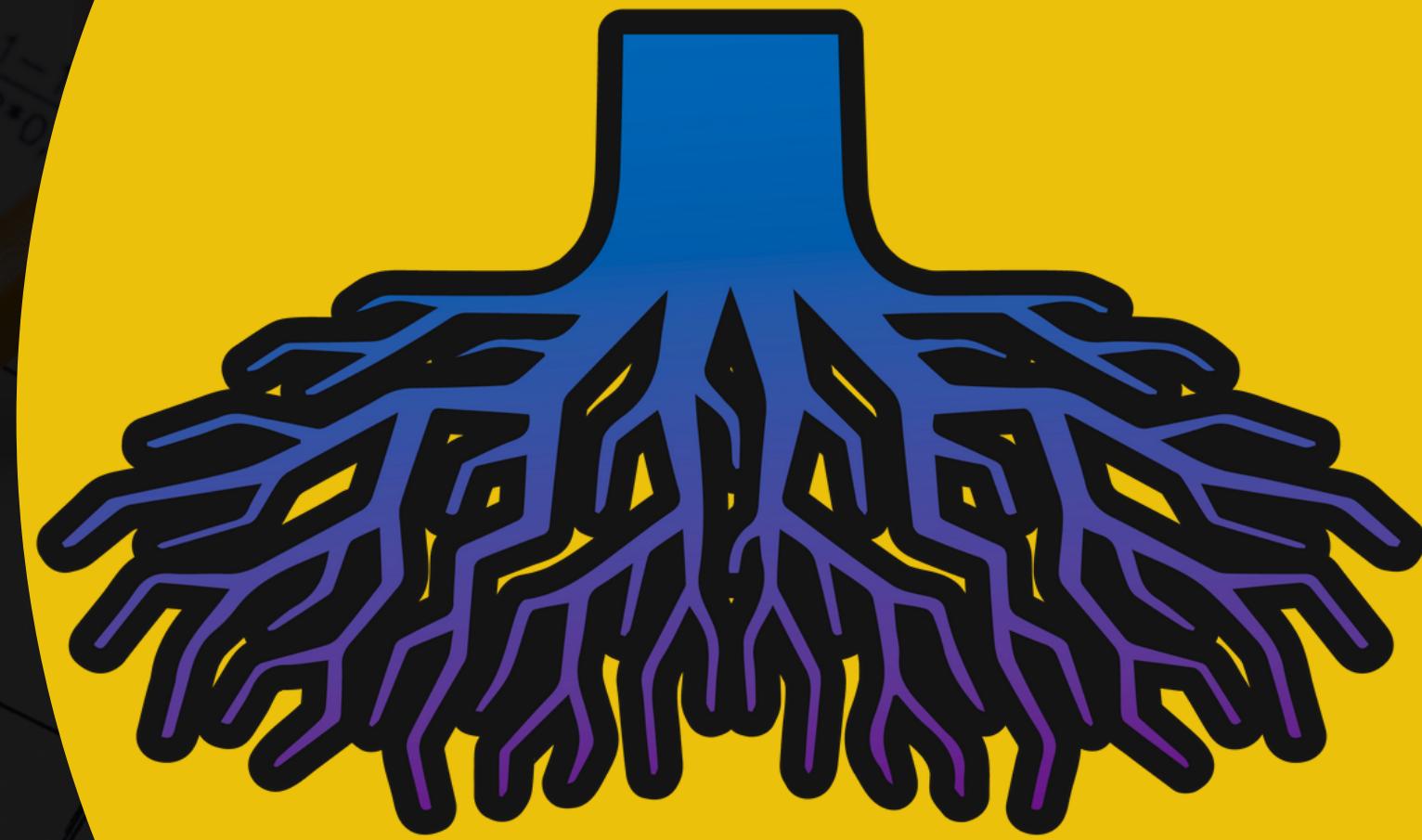
CREATED WITH YUML



CREATED WITH YUML

AULA COMPOSITE

RECURSÃO... RECURSÃO EVERYWHERE





OBJETIVOS

- APRESENTAR O PROBLEMA GERAL
- APRESENTAR UMA SOLUÇÃO UTILIZANDO O COMPOSITE
- CONHECER COMO A RECURSÃO AJUDA BASTANTE A NAVEGAR POR ESTE PADRÃO
- SER APRESENTADO PARA UMA BIBLIOTECA QUE UTILIZA DIVERSOS PADRÕES: SWING

PROBLEMAS

- COMO SERIA POSSÍVEL CRIAR UMA ESTRUTURA ONDE UMA UNIDADE E UM CONJUNTO DELAS POSSAM SER TRATADAS DE FORMA TRANSPARENTE, SEM DIFERENÇAS?



SOLUÇÃO

- DEFINIR UMA COMPOSIÇÃO QUE POSSA COMPORTAR TANTO UM ELEMENTO APENAS QUANDO UM CONJUNTO DESSES MESMOS ELEMENTOS

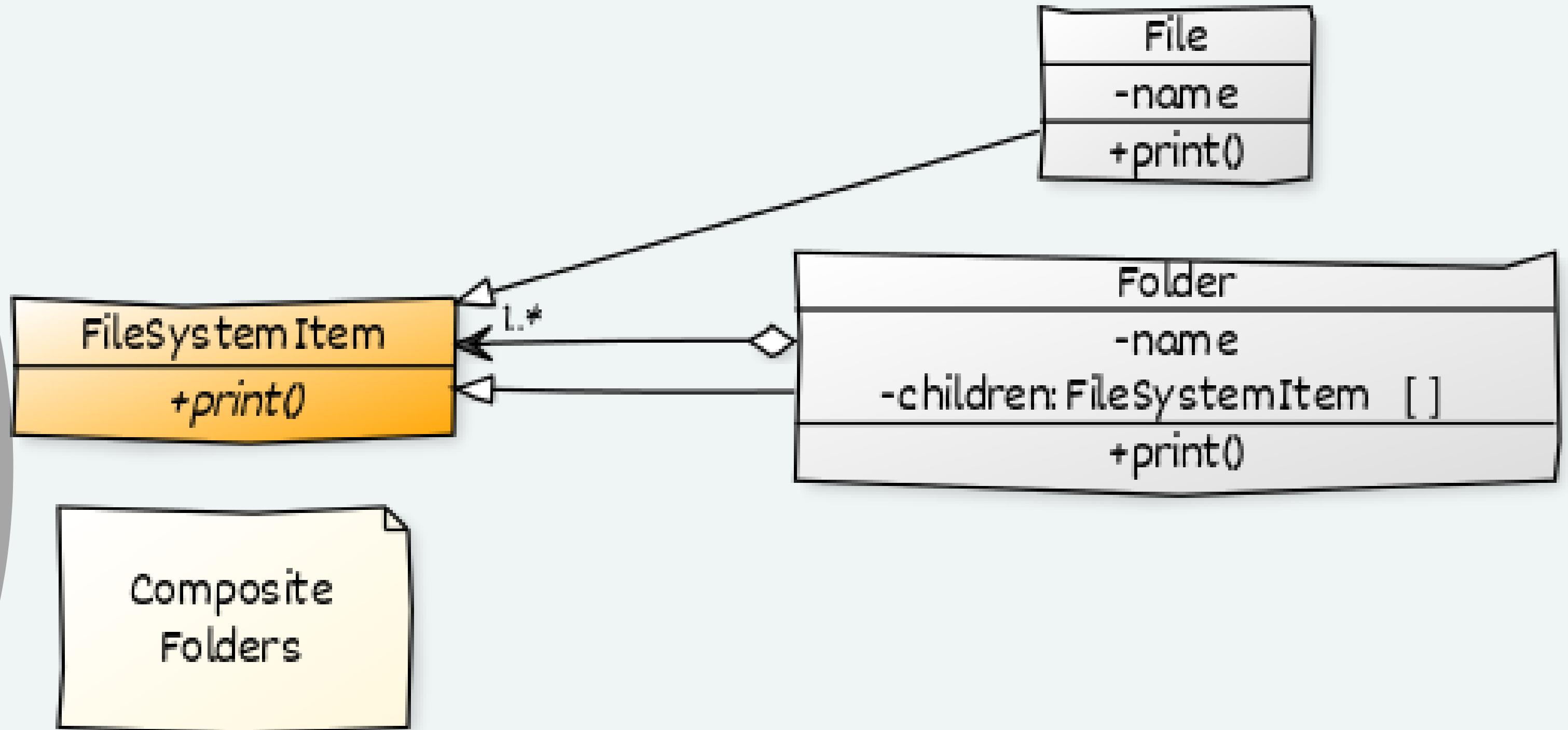




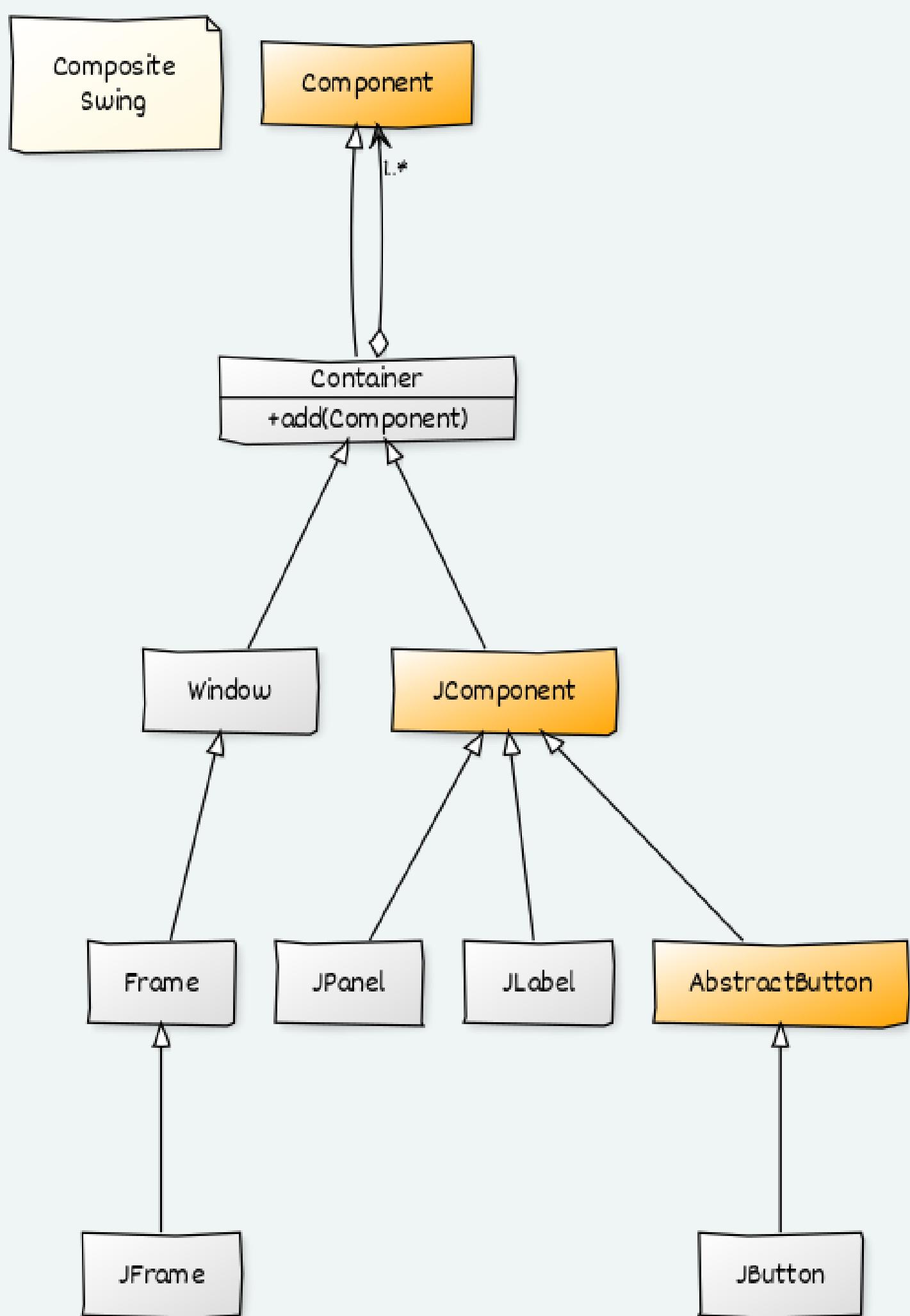
Compor objetos em estruturas de árvore para representar hierarquias todo-partes. Composite permite que clientes tratem objetos individuais e composições de objetos de maneira uniforme

GOF



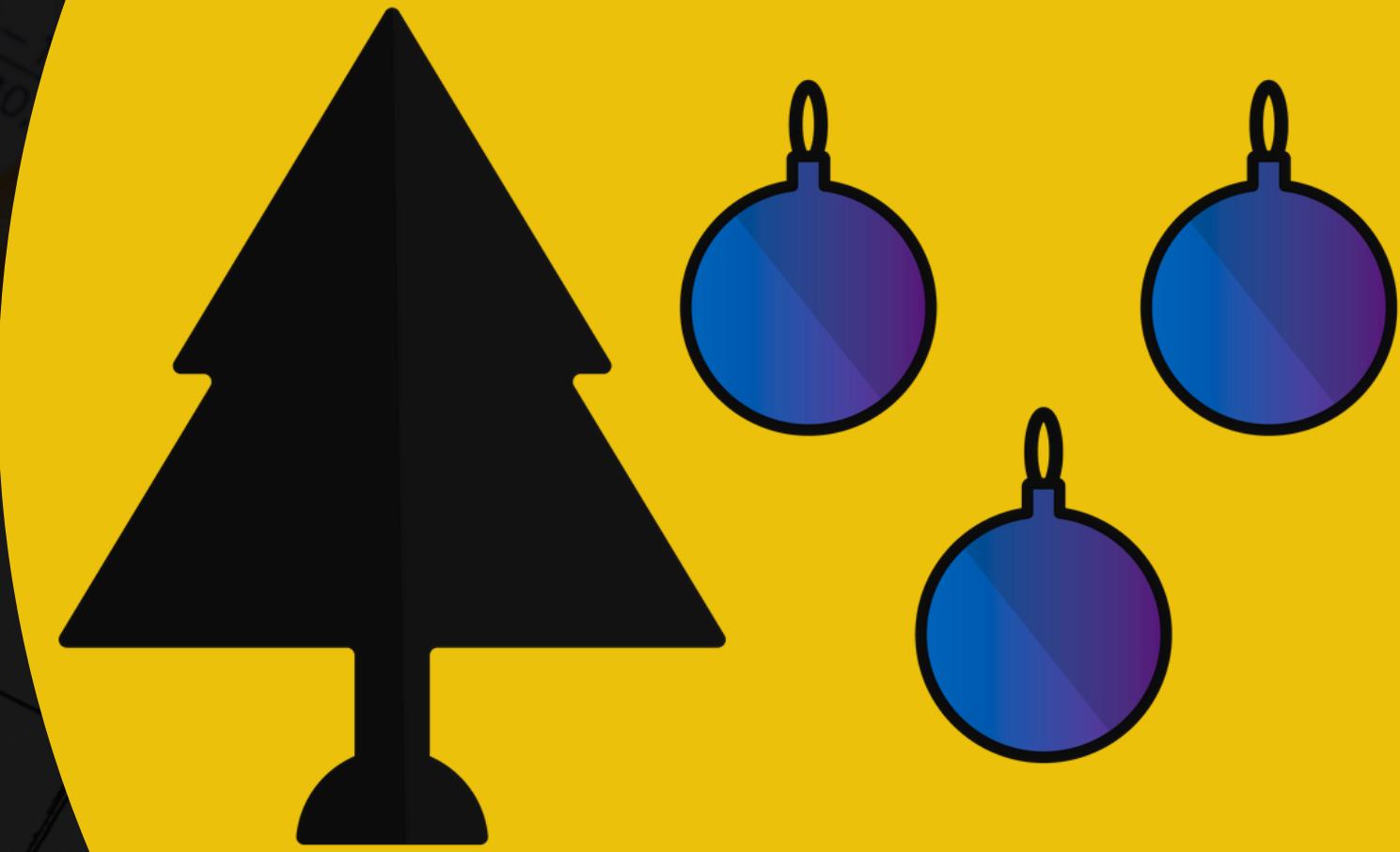


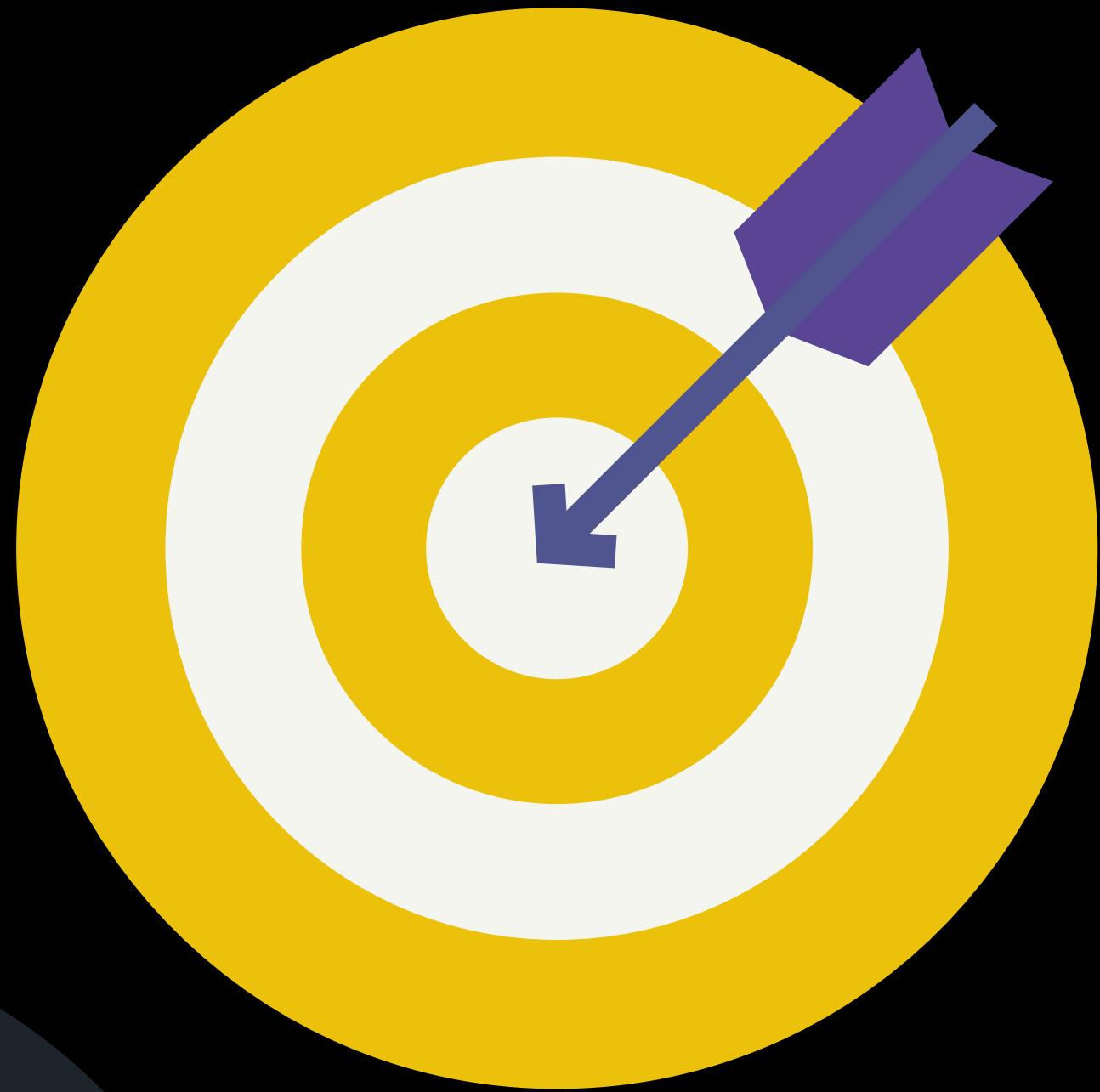
CREATED WITH YUML



AULA DECORATOR

VAMOS ADICIONAR SÓ MAIS UM
DETALHEZINHO?





OBJETIVOS

- APRESENTAR O PROBLEMA GERAL
- APRESENTAR UMA SOLUÇÃO UTILIZANDO O DECORATOR
- VER COMO O JAVA SWING UTILIZA ESTE PADRÃO
- APRENDER ALGUMAS BEBIDAS

PROBLEMAS

- COMO POSSO ADICIONAR
FUNCIONALIDADES DINAMICAMENTE EM
UM OBJETO?



SOLUÇÃO

- COM OBJETOS DECORATORS É POSSÍVEL ADICIONAR NOVAS RESPONSABILIDADES PARA UM OBJETO

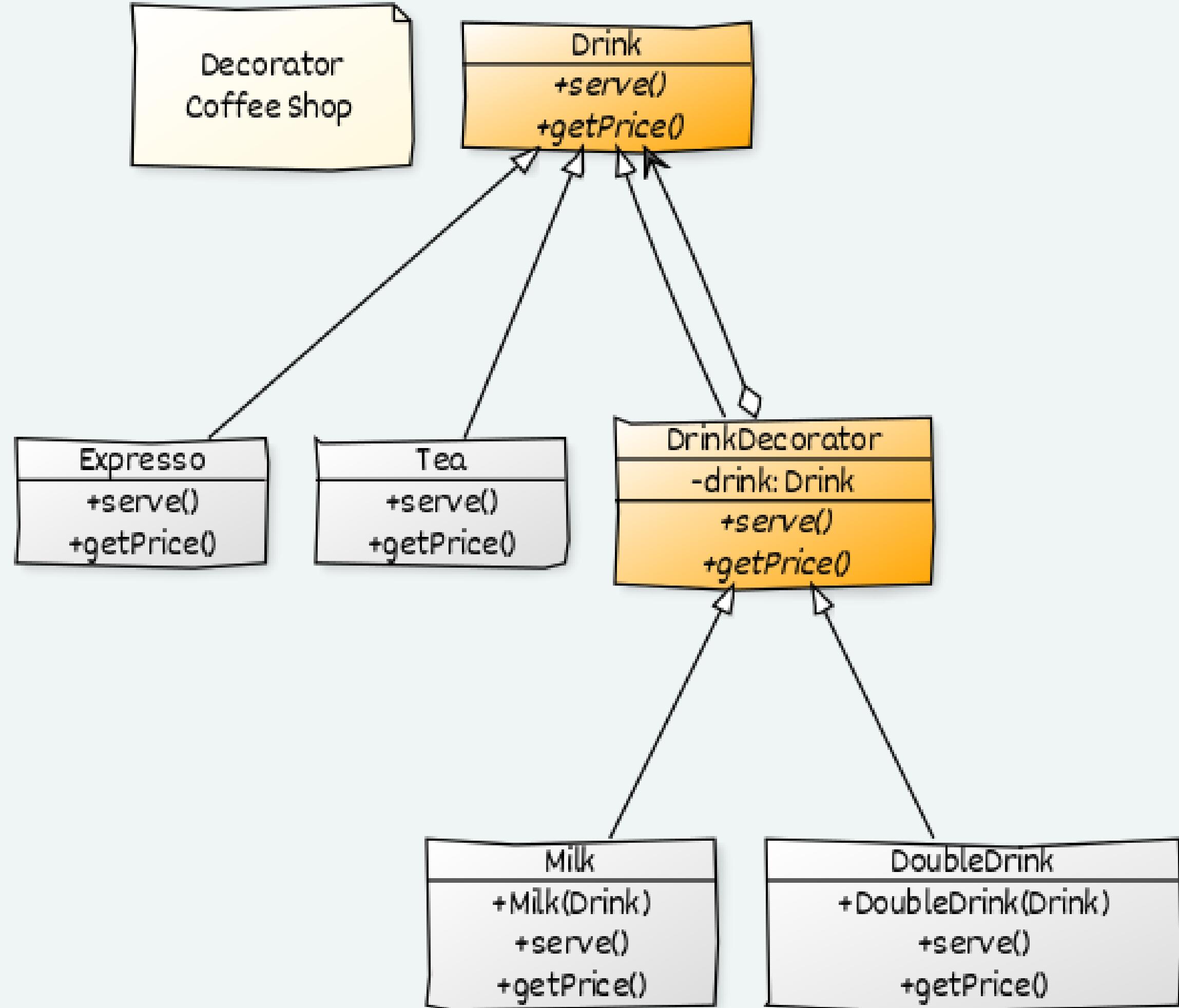


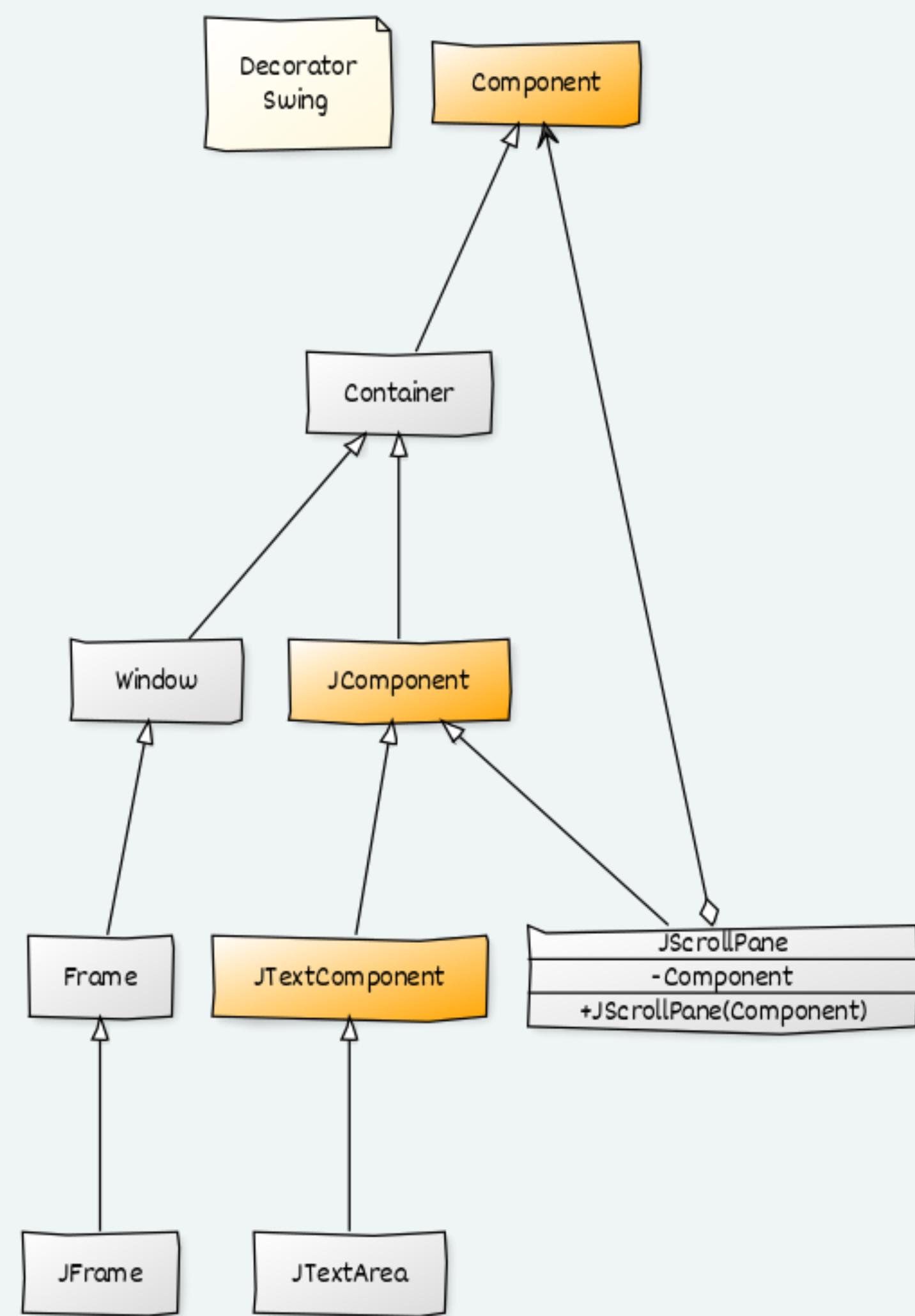
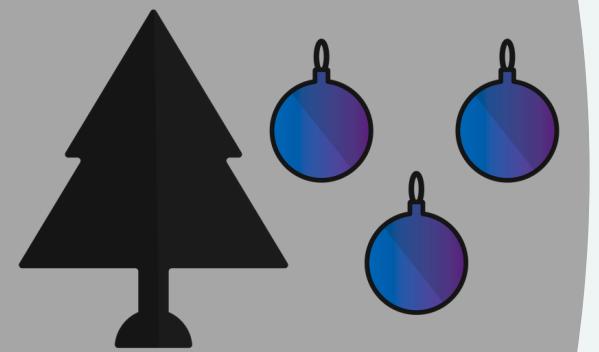


Anexar responsabilidades adicionais a um objeto dinamicamente. Decorators oferecem uma alternativa flexível ao uso de herança para estender uma funcionalidade.

GOF







AULA FAÇADE

AQUI ESTÁ O QUE VOCÊ PRECISA.
SÓ NÃO REPARA A BAGUNÇA ;)





OBJETIVOS

- APRESENTAR O PROBLEMA GERAL
- APRESENTAR UMA SOLUÇÃO UTILIZANDO O FACADE
- VER COMO PODEMOS FAZER VARIAÇÕES DEPENDENDO DO CONTEXTO

PROBLEMAS

- COMO POSSO SIMPLIFICAR O ACESSO A UM SISTEMA COMPLEXO UTILIZANDO UMA INTERFACE SIMPLES?
- COMO REDUZIR A COMPLEXIDADE INTERNA DE UM SISTEMA PARA O CLIENTE?



SOLUÇÃO

- DEFINIR UMA FACHADA QUE PROVISIONA UMA INTERFACE ÚNICA PARA UM CONJUNTO DE INTERFACES DE UM SISTEMA

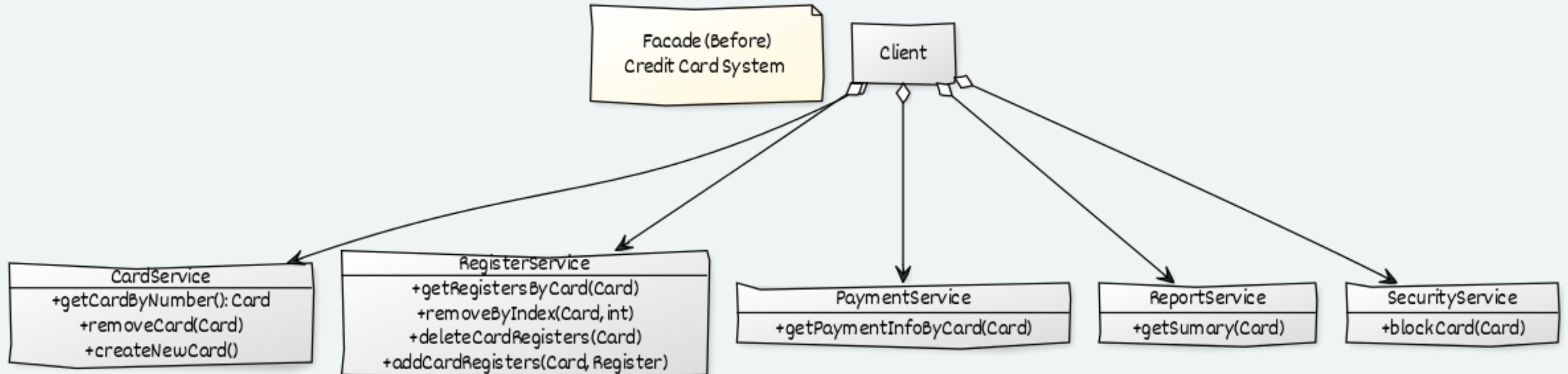




Oferecer uma interface única para um conjunto de interfaces de um subsistema. Façade define uma interface de nível mais elevado que torna o subsistema mais fácil de usar.

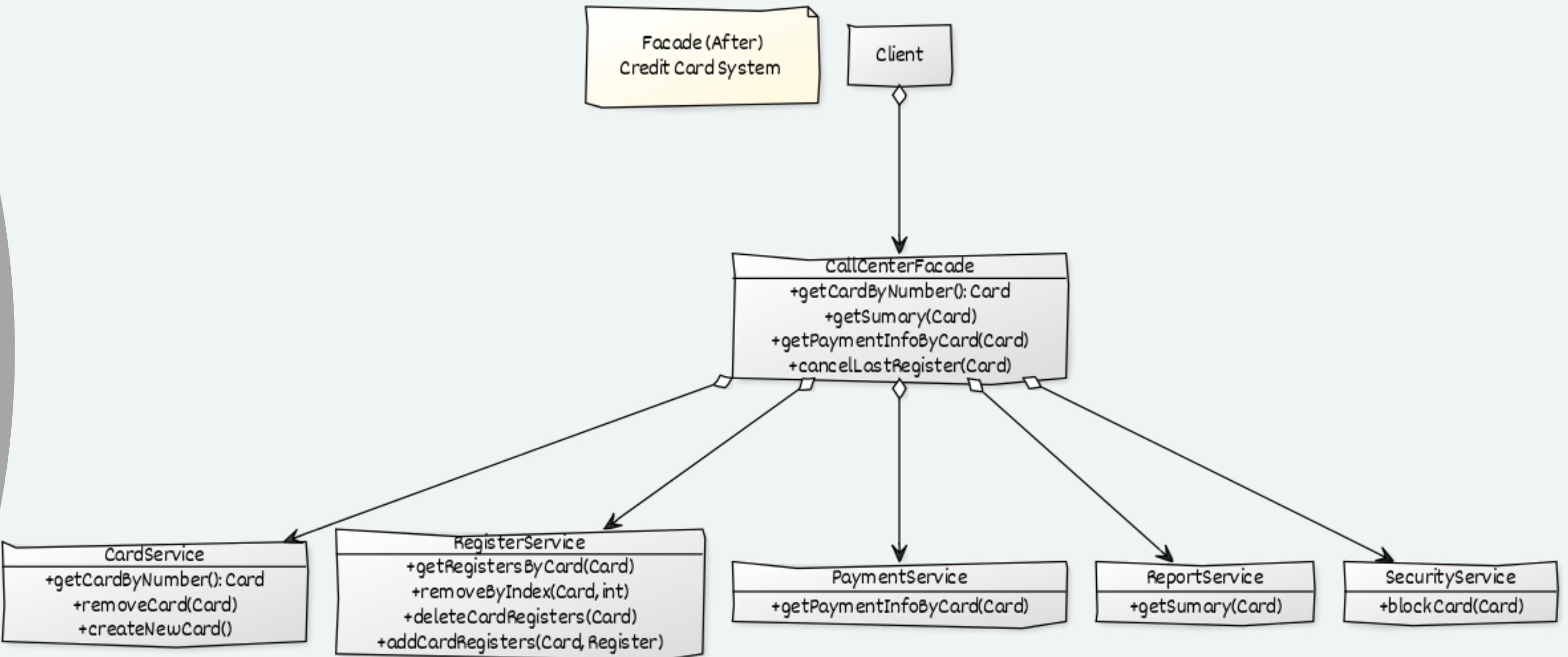
GOF





CREATED WITH YUML

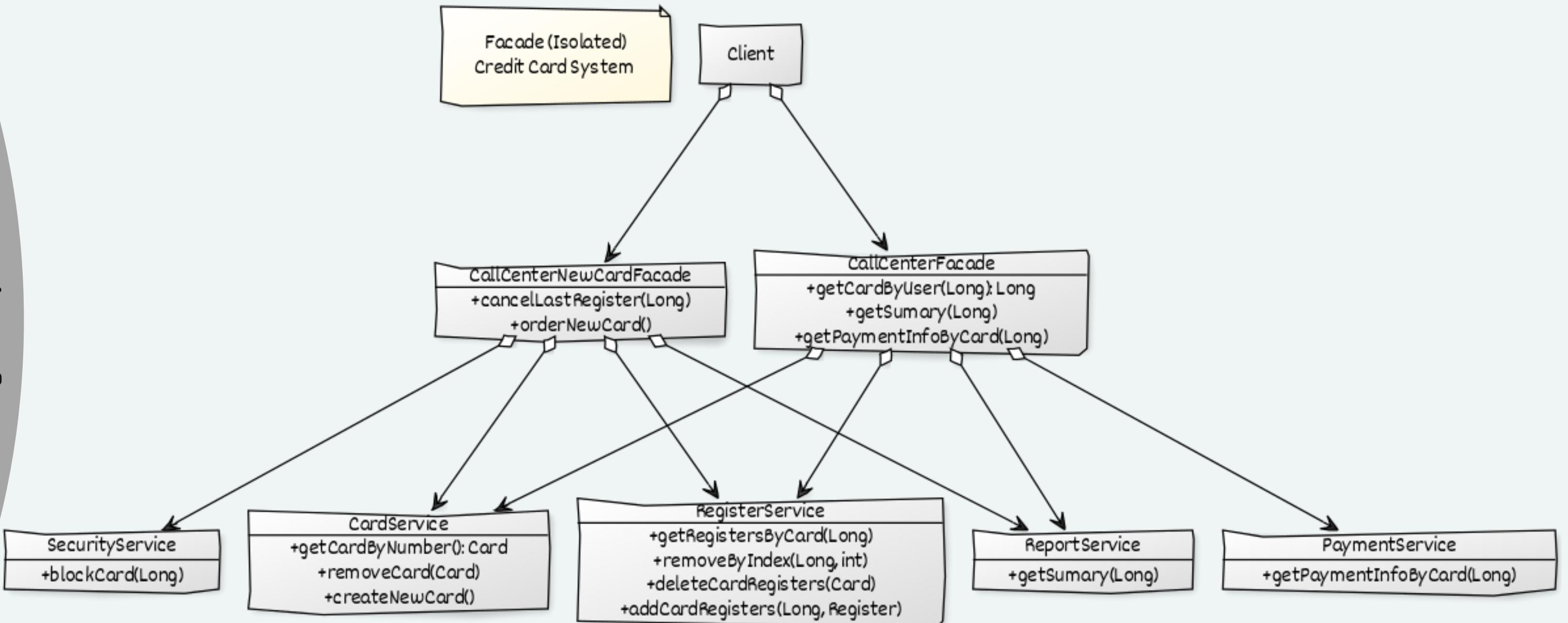




CREATED WITH YUML

Sample Code

```
public class Client {  
  
    public static void main(String[] args) {  
        CallCenterFacade facade = new CallCenterFacade();  
  
        // Using system objects  
        Card card = facade.getCardByUser(123456L);  
        facade.getSummary(card);  
  
        //Using simple values  
        Long cardNumber = 123456L;  
        facade.getSummary(cardNumber);  
  
    }  
}
```



CREATED WITH YUML

AULA FLYWEIGHT

É TIPO UM CACHE





OBJETIVOS

- APRESENTAR O PROBLEMA GERAL
- APRESENTAR UMA SOLUÇÃO UTILIZANDO O FLYWEIGHT

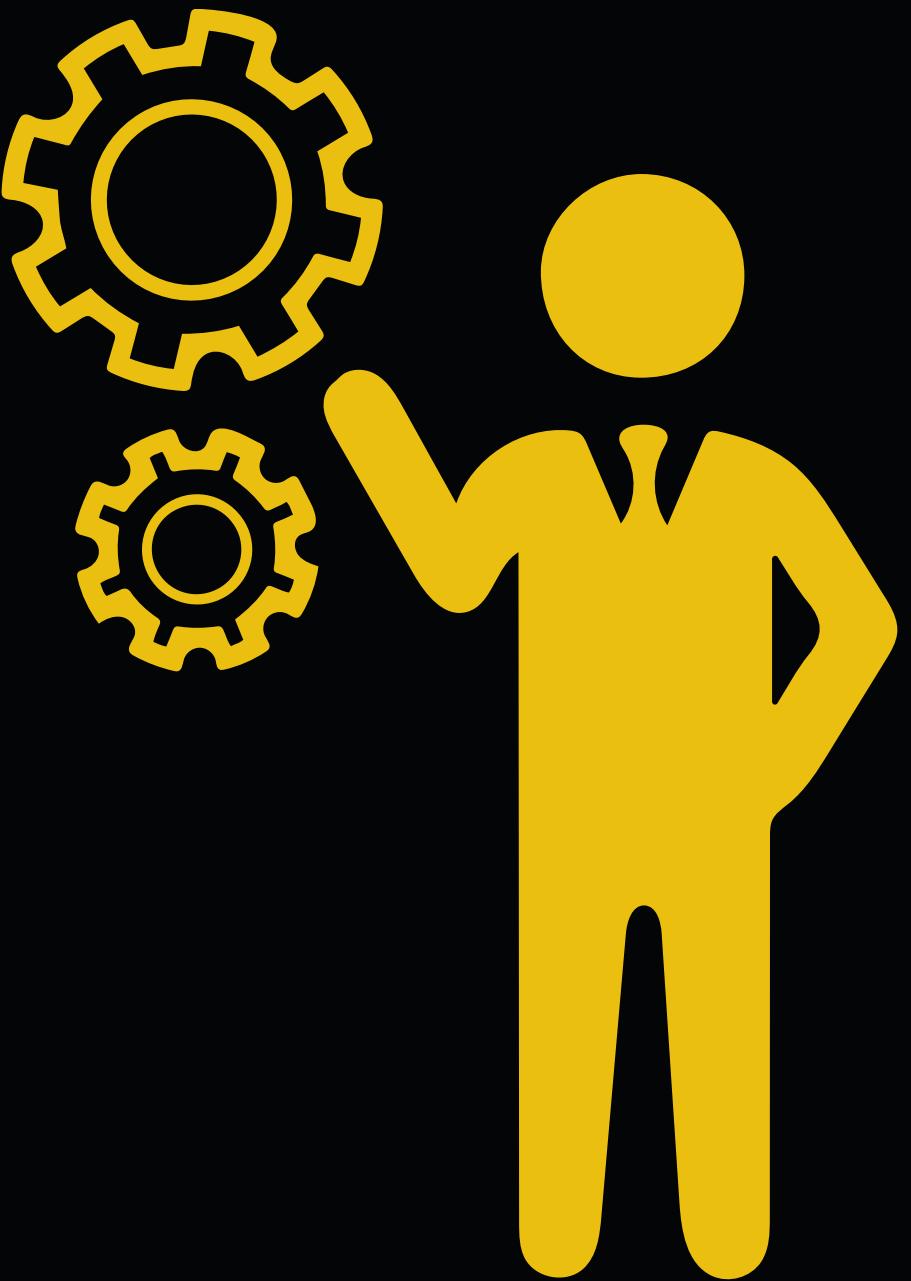
PROBLEMAS

- COMO POSSO TRABALHAR COM UMA
GRANDE QUANTIDADE DE OBJETOS NA
MEMÓRIA DE UMA FORMA MAIS
EFICIENTE?



SOLUÇÃO

- DIVIDIR O OBJETO ENTRE VALORES INTRÍNSECOS E EXTRÍNSECOS...
- ... CRIAR UM FLYWEIGHT QUE ARMAZENE OS VALORES INTRÍNSECOS...
- ... OS CLIENTES IRÃO COMPARTILHAR ESTES FLYWEIGHTS, ADICIONANDO OS VALORES EXTRÍNSECOS PONTUALMENTE

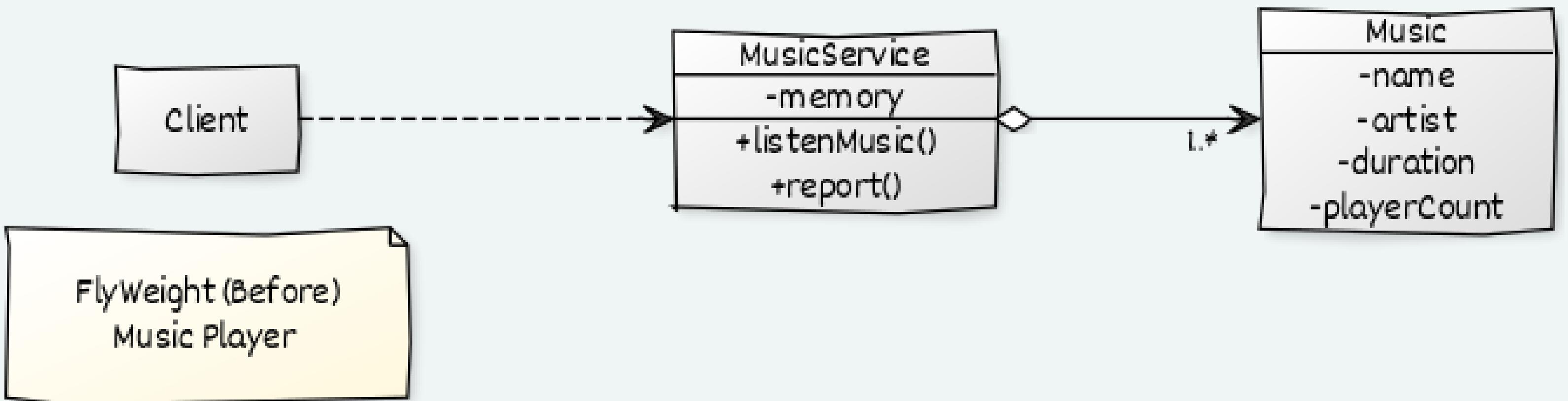




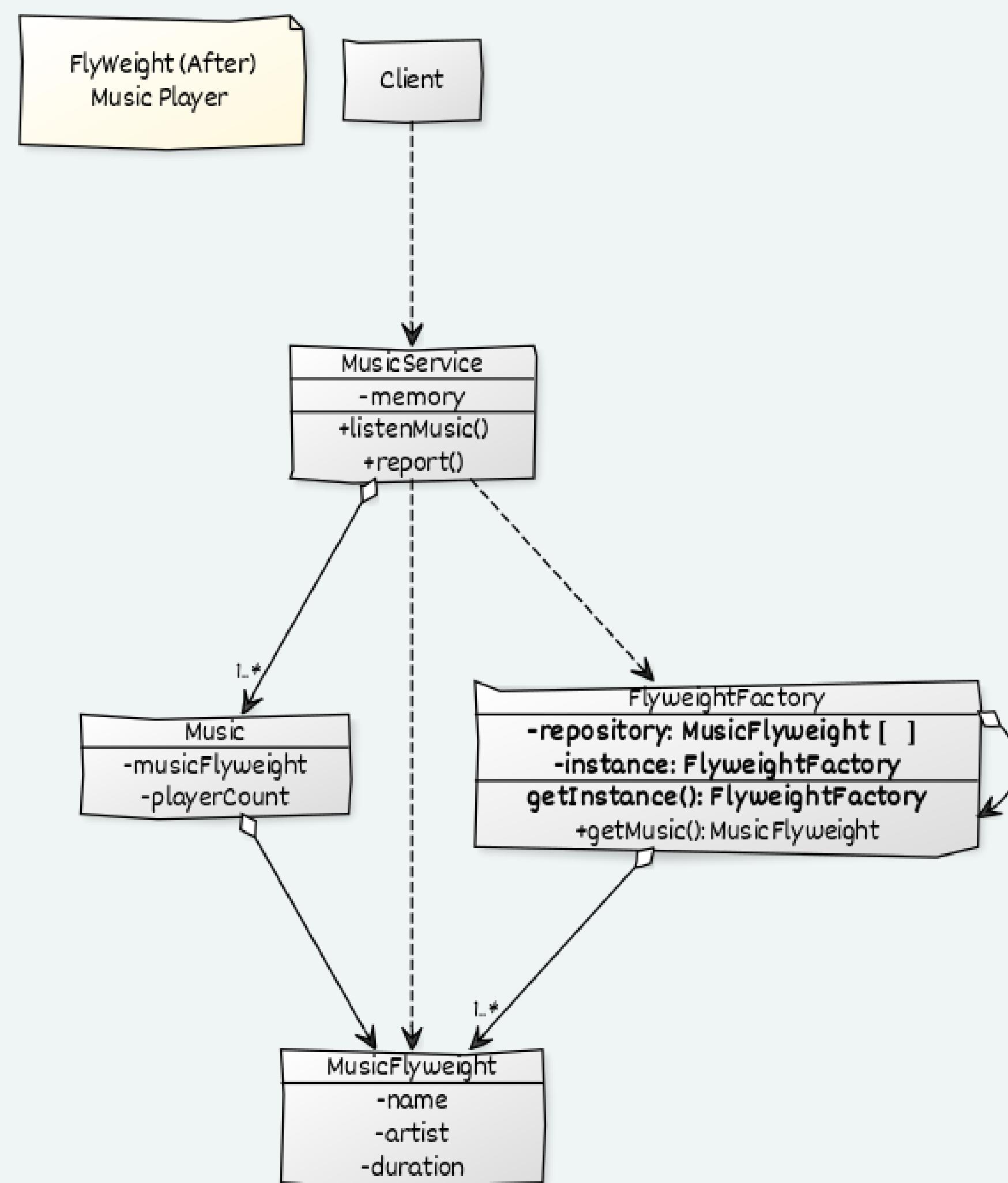
Usar compartilhamento para suportar grandes quantidades de objetos refinados eficientemente.

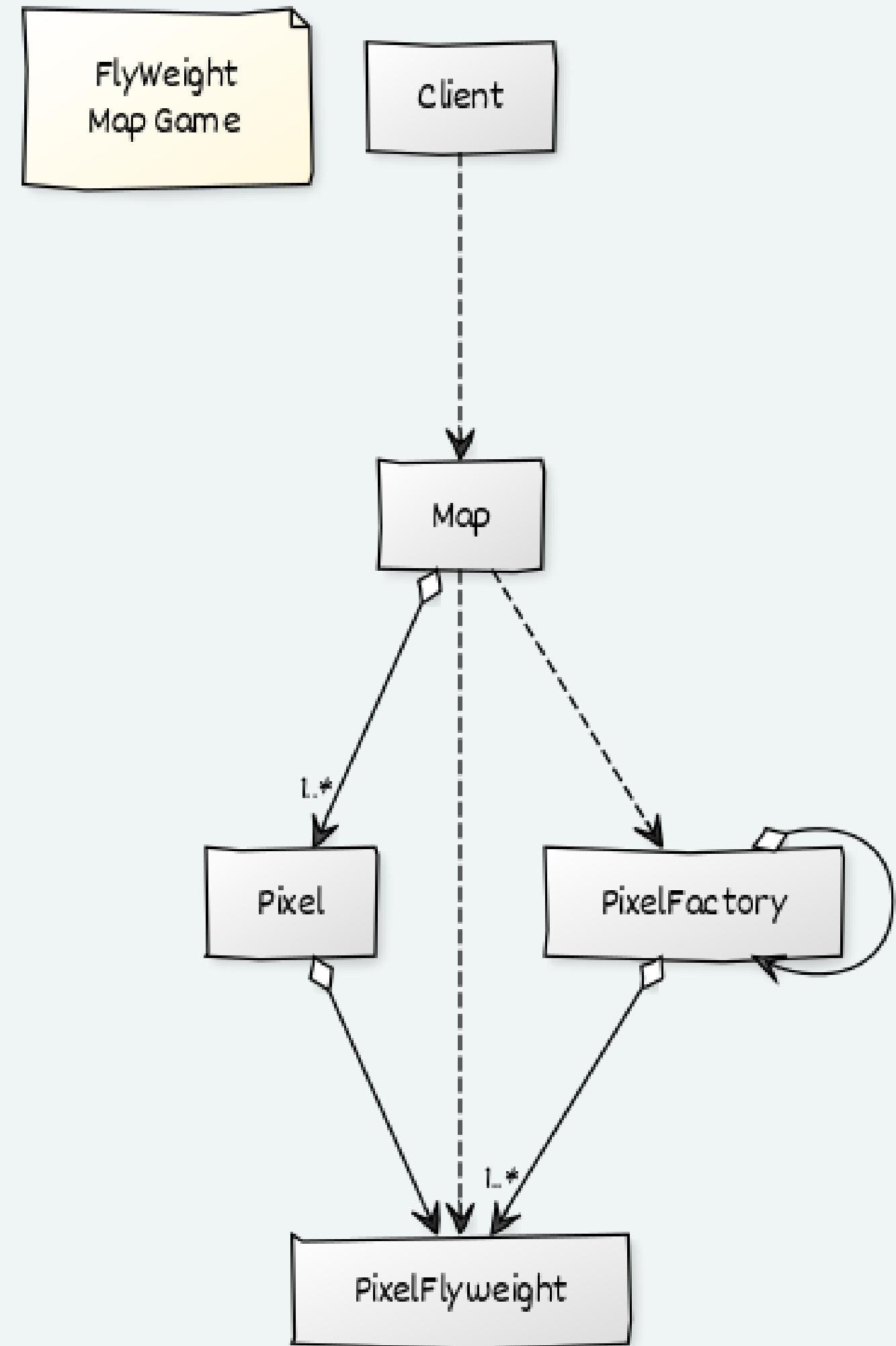
GOF





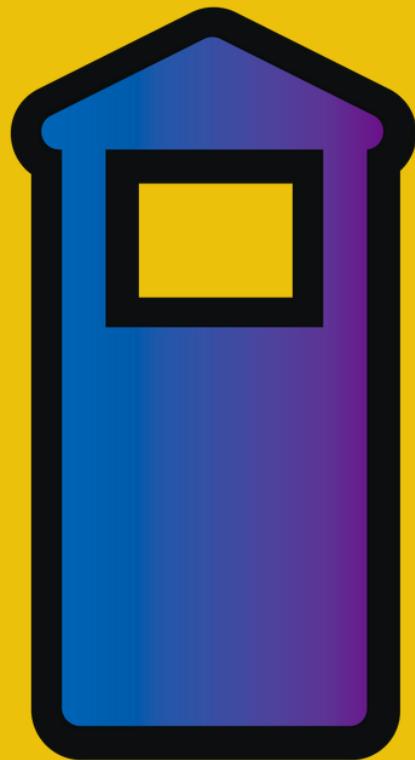
CREATED WITH YUML

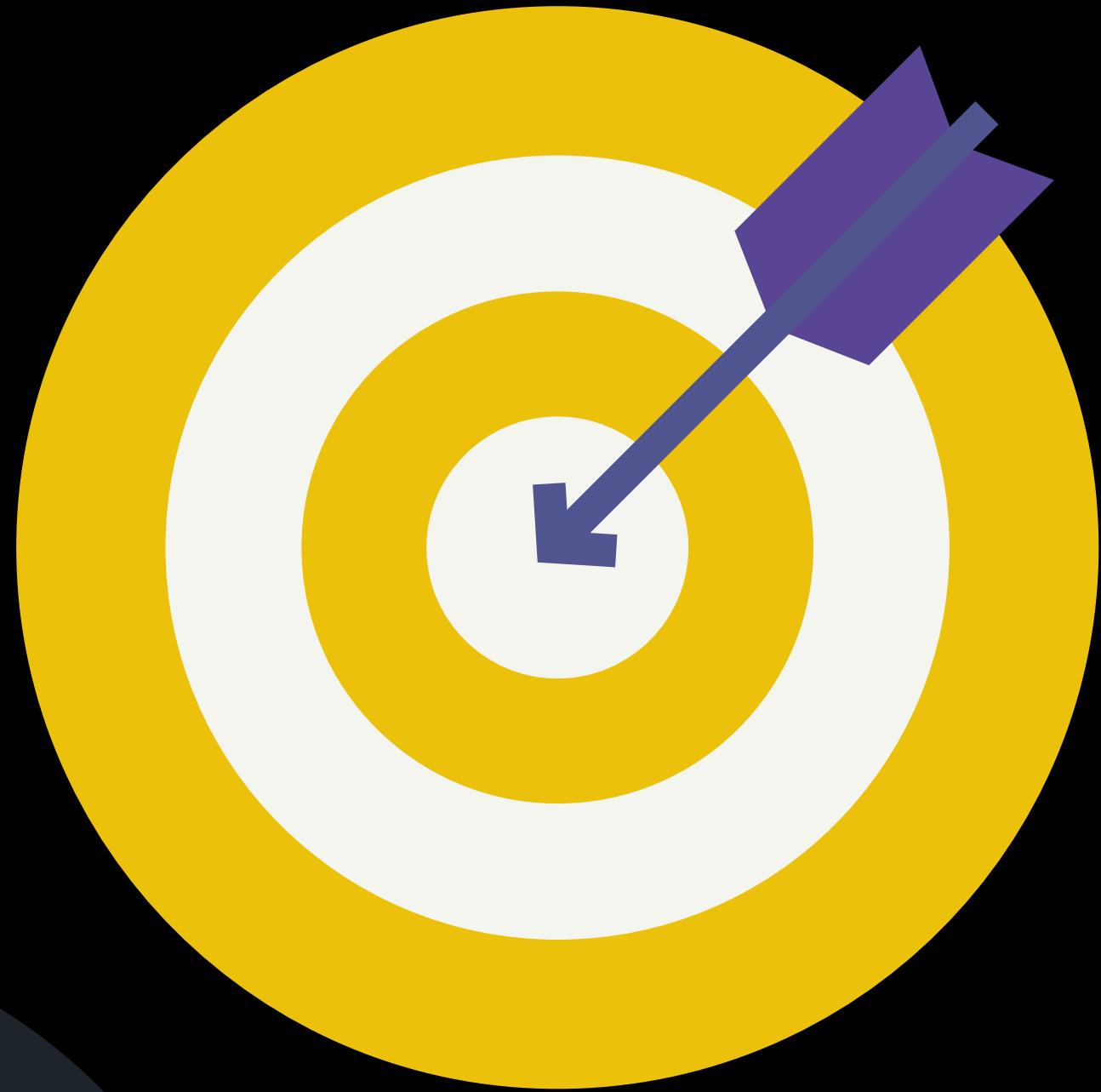




AULA PROXY

EU SOU UM PROXY PARA O SEU
CONHECIMENTO





OBJETIVOS

- APRESENTAR O PROBLEMA GERAL
- APRESENTAR UMA SOLUÇÃO UTILIZANDO O PROXY
- CONHECER OS TIPOS DE PROXY EXISTENTES

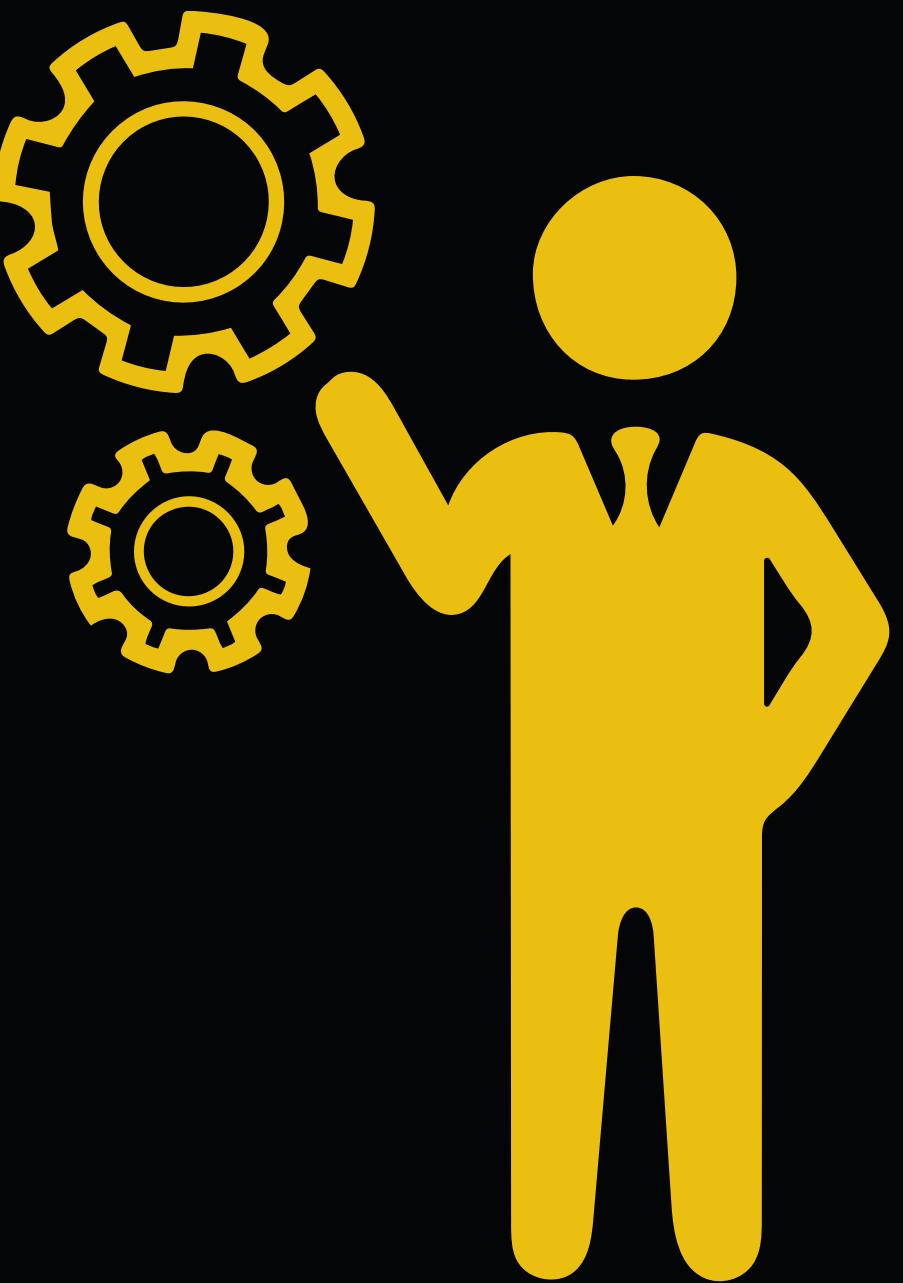
PROBLEMAS

- COMO POSSO CONTROLAR O ACESSO A ALGUM OBJETO?
- COMO POSSO ADICIONAR FUNCIONALIDADES DURANTE O ACESSO A ALGUM OBJETO?



SOLUÇÃO

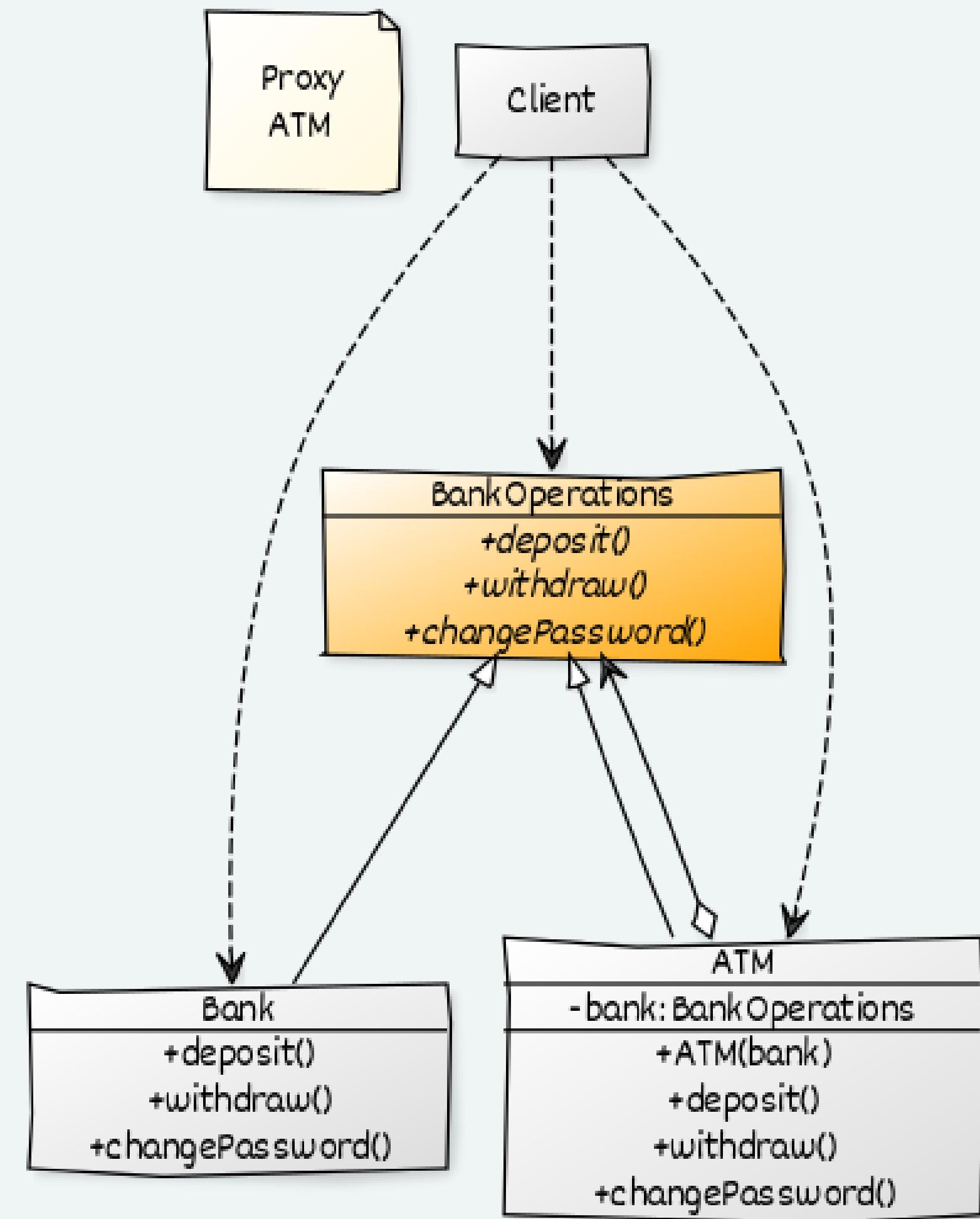
- UTILIZAR UMA ESTRUTURA DE PROXY
PARA ATUAR COMO SE FOSSE UM OUTRO
OBJETO
- ... UTILIZANDO O PROXY COMO FORMA DE
ACESSAR O OBJETO REAL, SERÁ POSSÍVEL
CONTROLAR O SEU ACESSO

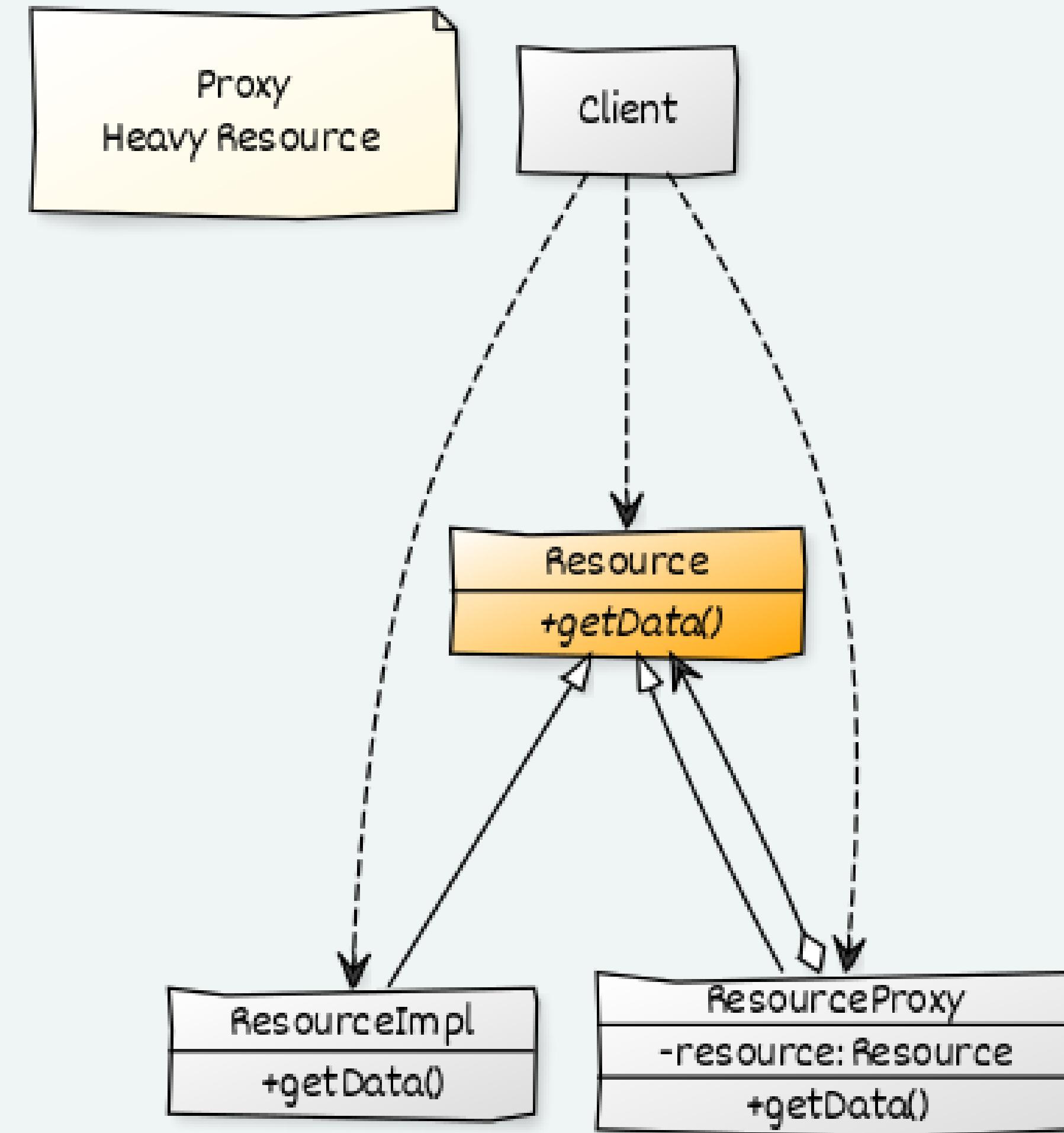


Prover um substituto ou ponto através do qual um objeto possa controlar o acesso a outro.

GOF







- Remote proxy
- Virtual proxy
- Protection proxy
- Smart Proxy

AULA MEDIATOR

APENAS UM GAROTO DE RECADOS





OBJETIVOS

- APRESENTAR O PROBLEMA GERAL
- APRESENTAR UMA SOLUÇÃO UTILIZANDO O MEDIATOR

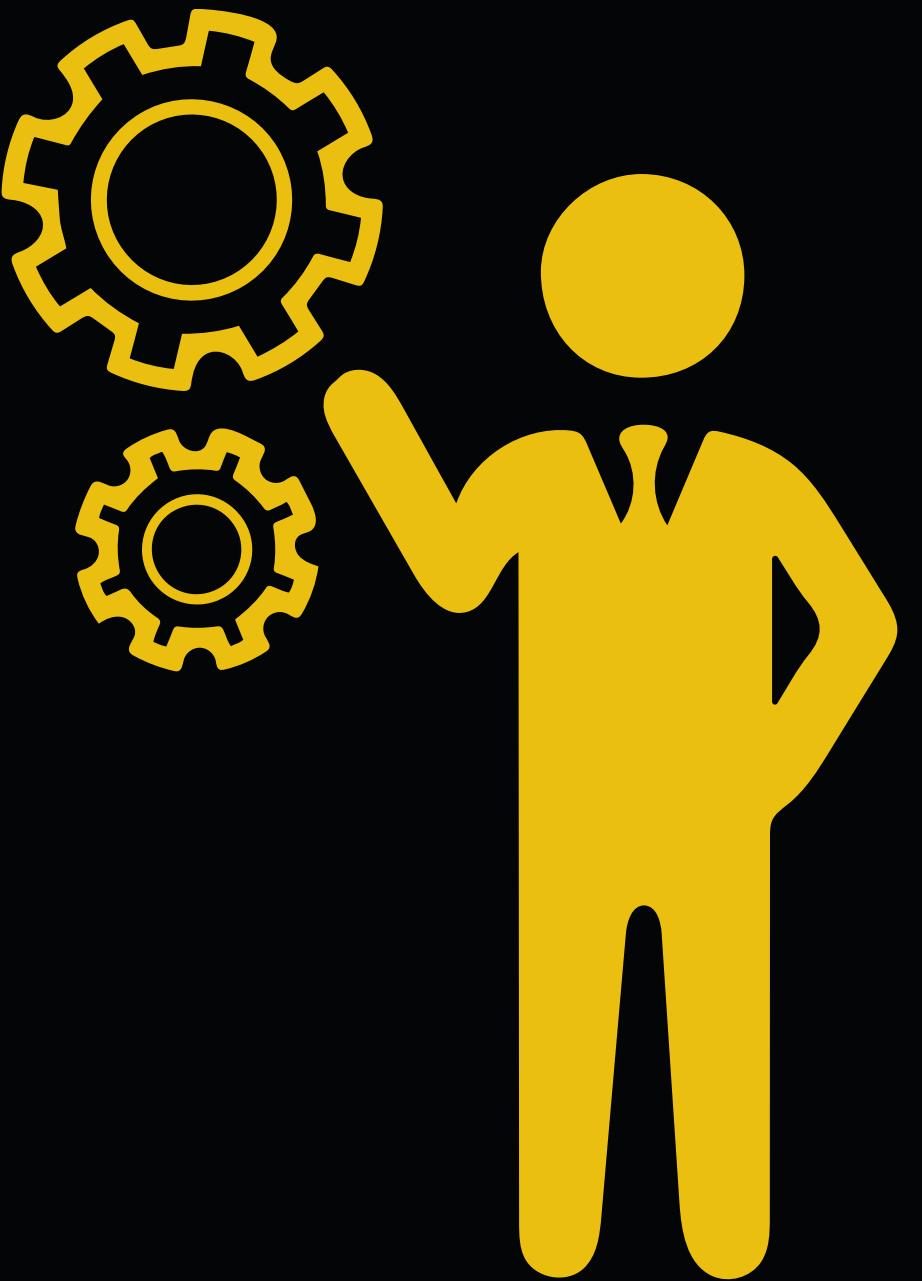
PROBLEMAS

- COMO POSSO REDUZIR O ACOPLAMENTO ENTRE OBJETOS QUE PRECISAM SE COMUNICAR?
- COMO POSSO DEIXAR A INTERAÇÃO ENTRE OBJETOS MAIS DINÂMICA?



SOLUÇÃO

- UTILIZAR UMA ESTRUTURA DE MEDIATOR PARA ENCAPSULAR A FORMA EM QUE OS OBJETOS SE COMUNICARÃO
- OS OBJETOS NÃO PRECISAM SE CONHECER, POIS TODA A INTERAÇÃO SERÁ REALIZADA ATRAVÉS DO MEDIATOR





Definir um objeto que encapsula como um conjunto de objetos interagem. Mediator promove acoplamento fraco ao manter objetos que não se referem um ao outro explicitamente, permitindo variar sua interação independentemente.

GOF



