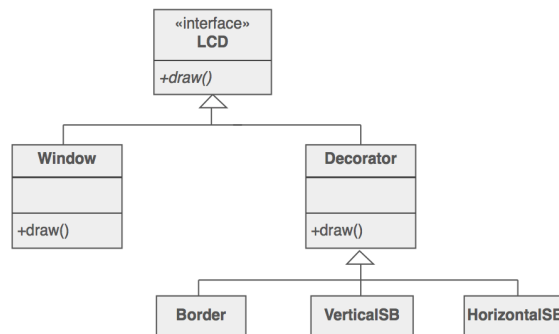


## O Padrão de Desenho Decorator

O padrão **Decorator** é um **padrão estrutural** que permite **adicionar novas responsabilidades (comportamento/estado)** aos **objetos em runtime**. Desta forma, a construção das características de um objeto solicitada pelo cliente dinamicamente.

Este padrão fornece uma alternativa à herança para estender a funcionalidade de uma classe. Para exemplificar o conceito, considere o diagrama seguinte.



No diagrama anterior, a classe *Window* representa o objeto principal do padrão, enquanto que a classe *Decorator* e derivadas representam a lógica adicional que pode ser adicionada ao objeto principal dinamicamente. Para exemplificar, o código seguinte adiciona os elementos necessários para criar uma estrutura composta pelo objeto principal *Window* e outros objetos utilizados para fazer a decoração de uma janela (*scrollbar* vertical, *scrollbar* horizontal e contorno).

```
Widget aWidget = new Border(new HorizontalSB(new VerticalSB( new Window( 80, 24 ))));
aWidget.draw();
```

### Exercício

Aplique o padrão Decorator a um componente responsável por fazer a **autenticação de um serviço**. Este componente deverá permitir **decorar a funcionalidade principal** com funcionalidades de:

- **logging**

- **verificação de palavras inseguras na password**, recorrendo a uma API REST que permite verificar se a palavra utilizada durante o processo de autenticação é insegura (faz parte do dicionário).

#### Specification

Documentação [JavaDocs](#)

#### Deadline

2020-03-20 23:55

[Submit for Testing](#)[Submit for Evaluation](#)