



TÉCNICAS DE PROGRAMAÇÃO – Checkpoint #3

EQUIPE:

ABNER DE LIMA – 398067

ANDERSON NOBRE – 352132

EDEALCIA REGINA – 372164

FCO TIAGO FONTENELE – 393944

THIAGO FRAXE – 397796

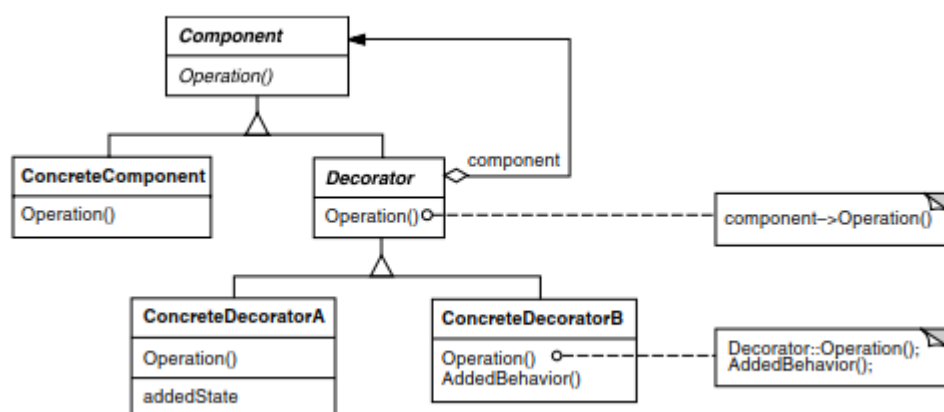
DESCRIÇÃO DOS PADRÕES DE PROJETO IMPLEMENTADOS

Foram utilizados na implementação do nosso projeto **Padrões Estruturais, Comportamentais e Arquiteturais**.

ESTRUTURAL

- Decorator

ESTRUTURA PADRÃO DECORATOR



Equivalência das classes usadas no nosso projeto:

package view.janelas

JanelaAbstrata => Component

Classe raiz do padrão de projeto decorator para janelas. Essa classe mantém o JFrame da janela que está sendo exibida.

Janela => ConcreteComponent

Classe concreta de Janela no padrão de projeto Decorator.

DecoradorJanela => Decorator

Classe abstrata que define o decorador de janela. Essa classe mantém um objeto com a janela que está sendo exibida.

JanelaMenu => ConcreteDecoratorA

Classe de decoração de janela. Adiciona a barra de menu no topo da janela.

JanelaNavega => ConcreteDecoratorB

Classe de decoração de janela. Adiciona a possibilidade de navegar para uma página.

package view.paginas;

PaginaAbstrata => Component

Define uma página abstrata para ser decorada.

Pagina => ConcreteComponent

Classe concreta que representa uma página em branco da aplicação.

DecoradorPagina => Decorator

Classe abstrata do padrão de projeto decorador para páginas.

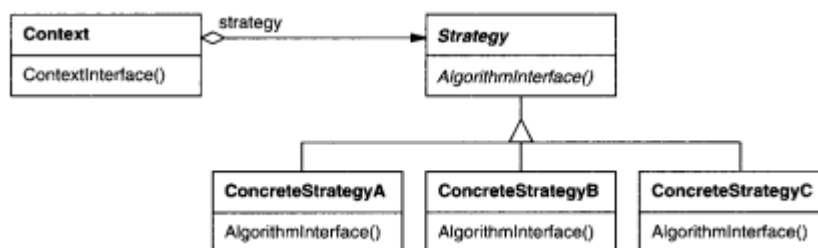
PaginaInicial => ConcreteDecorator

Decorador de página que carrega o visual da página inicial do programa.

COMPORTAMENTAL

- Strategy

ESTRUTURA PADRÃO STRATEGY



Equivalência das classes usadas no nosso projeto:

package model;

Analises => Context

Mantém referência com os objetos *AnaliseUnivariada* e *AnaliseMultivariada*(Strategy).

AnaliseUnivariada; AnaliseMultivariada => Strategy

Cada uma dessas duas classes funciona como um compositor e definem uma interface comum para as diversas métricas estabelecidas no projeto (univariadas e multivariadas respectivamente)

Media; Moda; Mediana; Minimo; Maximo; Variancia; DesvioPadrao; Skewness; Kurtosis; Covariancia; Pearson => ConcreteStrategy

Cada uma dessas classes implementa o algoritmo específico de cada métrica usando as interfaces *AnaliseUnivariada* e *AnaliseMultivariada*.

package model;

Graficos => Context

Mantém referência com os objetos *GraficosQuantitativos* e *GraficosQualitativos*(Strategy).

GraficosQuantitativos; GraficosQualitativos => Strategy

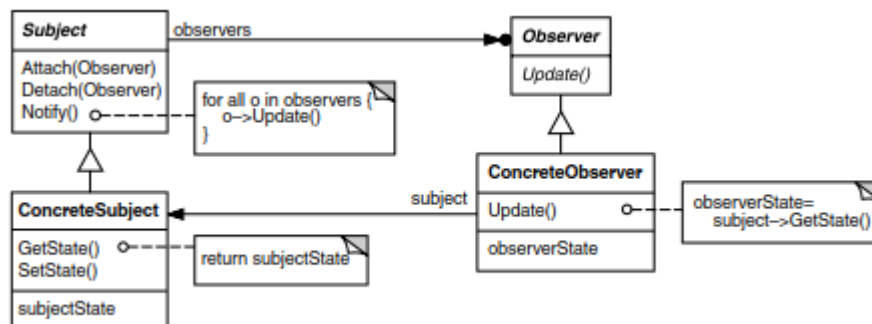
Cada uma dessas duas classes funciona como um compositor e definem uma interface comum para os tipos de apresentação gráfica estabelecidos no projeto (Histograma, TabelaFrequencias, Boxplot, Scatterplot, TabelaContigencia, TabelaFrequencias, GraficosDeBarras)

Histograma; TabelaFrequencias; Boxplot; Scatterplot; TabelaContigencia; TabelaFrequencias; GraficosDeBarras => ConcreteStrategy

Cada uma dessas classes implementa o algoritmo específico de apresentação gráfica dos dados usando as interfaces *GraficosQuantitativos* e *GraficosQualitativos*.

- **Observer**

ESTRUTURA PADRÃO OBSERVER



Equivalência das classes usadas no nosso projeto:

package view.util;

MenuObservavel => Subject

Fornece a interface para um objeto de Menu observável.

MenuObservador => Observer

Fornece a interface de atualização para um objeto de Menu observador.

package view.janelas;

JanelaMenuListener => ConcreteSubject

Classe concreta que é observada. Processa a escolha do item da barra de menu de uma janela. Faz uso do padrão de projeto observer pois implementa a interface MenuObservavel.

package gerenciamento;

MenuBarListener => ConcreteObserver

Essa classe é responsável por observar a seleção do item da barra de menu. Trabalha com a classe JanelaMenuListener.

package model;

TabelaObservable => Subject

Fornece a interface para um objeto de Tabela observável.

Tabela => ConcreteSubject

Classe concreta que é observada. Faz uso do padrão de projeto observer pois implementa a interface TabelaObservable.

TabelaObserver => Observer

Fornece a interface de atualização para um objeto de Tabela observador.

package view.paginas;

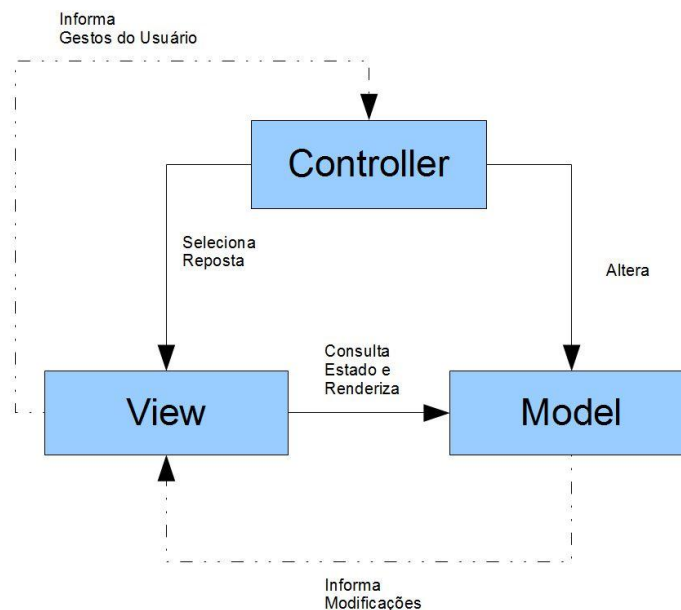
Paginainicial => ConcreteObserver

Classe concreta que também é usada como observador. Trabalha com a classe Tabela.

ARQUITETURAL

- **MVC (Model-View-Controller)**

ESTRUTURA PADRÃO MVC



Organização dos pacotes e classes de acordo com o padrão:

- **MODEL**
 - **tabela**
 - Tabela.java
 - TabelaObservable.java
 - TabelaObserver.java
 - **univariada**
 - Media.java
 - Moda.java
 - Mediana.java
 - Minimo.java
 - Maximo.java
 - Variancia.java
 - DesvioPadrao.java
 - Skewness.java
 - Kurtosis.java
 - **multivariada**
 - Covariancia.java
 - Pearson .java
 - **graficos**
 - Graficos.java
 - GraficosQuantitativos.java
 - GraficosQualitativos.java
 - Histograma.java
 - TabelaFrequencias.java
 - Boxplot.java
 - Scatterplot.java
 - TabelaContigencia.java
 - TabelaFrequencias.java
 - GraficosDeBarras.java
- **VIEW**
 - **janelas**
 - DecoradorJanela.java
 - Janela.java
 - JanelaAbstrata.java
 - JanelaMenu.java
 - JanelaMenuListener.java
 - JanelaNavega.java
 - **paginas**
 - DecoradorPagina.java
 - MultivariadaListener.java
 - Pagina.java
 - PaginaAbstrata.java
 - PaginaInicial.java
 - QualitativosListener.java
 - QuantitativosListener.java
 - UnivariadaListener.java

- **util**
 - Botaolcone.java
 - BotaoRadio.java
 - Coluna.java
 - Cores.java
 - MenuObservador.java
 - MenuObservavel.java
 - Titulo.java

- **CONTROLLER**

- AbrirItemController.java
- DesenvolvedoresItemController.java
- FecharItemController.java
- ItemController.java
- SalvarItemController.java
- SobreItemController.java
- SuporteItemController.java