

ATIVIDADE - PADRÃO OBSERVER

Instruções:

- Esta atividade deve ser realizada considerando o mesmo grupo de Seminário de Sistemas Distribuídos;
- A atividade contará como a segunda nota da etapa 2 e valerá 100 pontos;
- Deverá ser entregue até o dia 28/12/2020.

Atividades:

1. Pense em como o padrão Observer pode ser aplicado no seu projeto de Sistemas Distribuídos;
 - a. Discorra sobre o problema que o padrão está resolvendo no seu projeto;
 - b. Desenvolva o diagrama de classes representando a aplicação do padrão;
 - c. Implemente na linguagem definida para o seu projeto, mesmo que de forma simplista, o padrão;
2. Envie o projeto para o Github e envie o link para esta atividade no Sala de Aula, juntamente com os arquivos dos diagramas em .pdf ou .png.

Observação:

Caso o padrão não se aplique no seu projeto: i) Explique com os fundamentos do padrão o porque do mesmo não se aplicar ao seu projeto; ii) Realize a atividade considerando um ou dois exemplos de projeto (implementado(s) em java); iii) Envie na atividade a descrição do(s) exemplo(s).

Padrão Observer

1. Por que não se aplica no projeto?

Por que ele tem como objetivo permitir que objetos interessados sejam avisados da mudança de estado ou outros eventos ocorrendo em um outro objeto. Ele permite a criação de dois atores principais, o primeiro é o objeto que está sendo observado e o segundo é o objeto observador, os dois possuem interfaces para o registro de métodos obrigatórios. Tem um tipo de relacionamento de dependência um-para-muitos entre os mesmos objetos, dessa forma quando o estado do objeto for alterado, todos os objetos dependentes recebem uma atualização.

Assim , esse padrão deve ser utilizado quando uma abstração tem dois aspectos, um dependente do outro. Encapsulando esses aspectos em objetos separados, permitindo variá-los e reutilizá-los independentemente, ou quando uma mudança em um objeto exige mudanças em outros, e não sabe-se quantos objetos necessitam ser mudados, ou ainda quando um

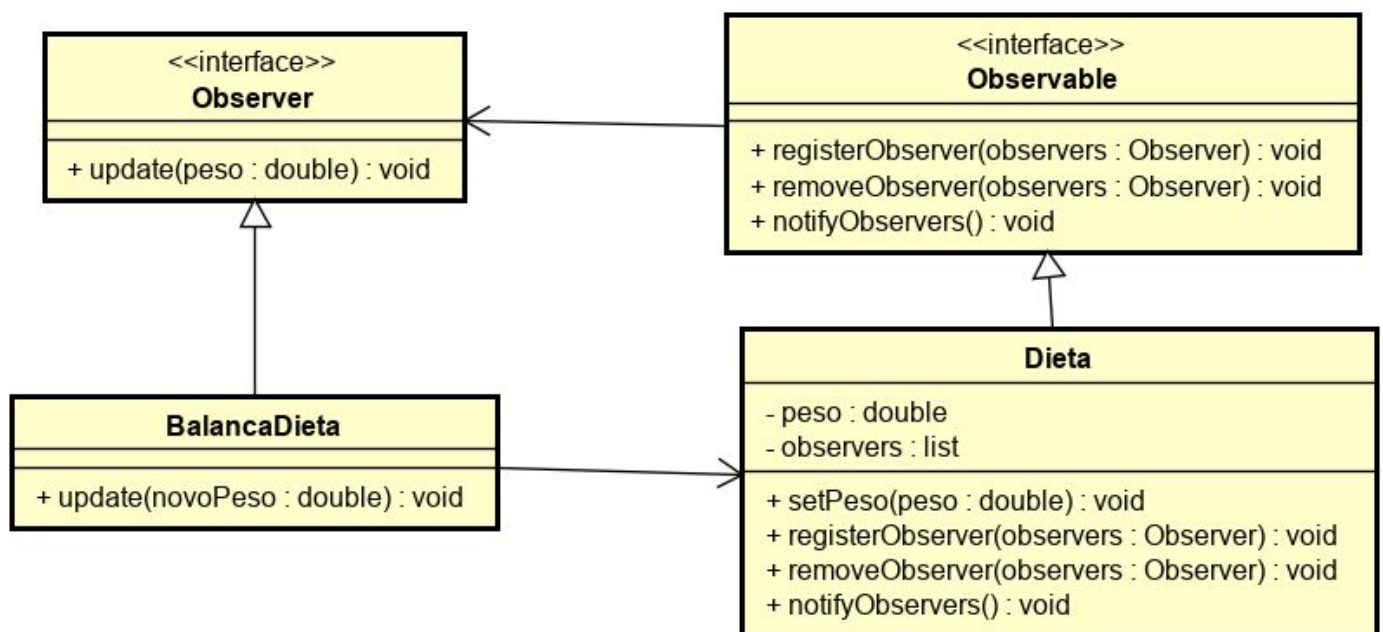
objeto deveria ser capaz de notificar outros objetos sem fazer hipóteses, usar informações, sobre quem são esses objetos. Em outras palavras, não é desejado que esses objetos sejam fortemente acoplados.

Desta forma, nenhuma dessas situações foi identificada no nosso projeto para que houvesse a aplicação do padrão no mesmo.

2. Que problema que o padrão está resolvendo neste projeto?

O exemplo a seguir mostra um sistema que gerencia dietas, onde o paciente informa diariamente as alterações ocorridas com seu peso e o sistema apresenta notificação informando se o peso ultrapassou ou se está no limite estabelecido.

Para isso, a solução contém uma classe que recebe essa informação e que atrela à ela mesma uma outra classe (O observer), para assim notificar toda alteração que aconteça. Quando a mesma ultrapassar o limite pré definido pelo IMC de massa corpórea o observer será avisado e então tomará as medidas cabíveis.



É possível observar que a interface **observer** possui o método **update(double peso)** que apresenta para cada observador uma regra específica de atualização. Desta forma, a interface estabelece um contrato obrigatório de implementação para os objetos observadores, ao qual terão que realizar alguma ação, com as notificações de mudança no objeto observado.

As classes **dieta** e **balança dieta** representam o processo necessário para ouvir os eventos de mudança no peso da dieta e notificar os respectivos **observers** que irão validar tal mudança.

