

2.3.1 Padrão Observer.

Temos um padrão que mantém seus objetos atualizados quando algo importante ocorre. O objeto pode até decidir no tempo de execução se deseja ser mantido informado. O Padrão Observer é um dos mais usados no JDK e é incrivelmente útil. Antes de terminar, também iremos analisar relações um-para-muitos e ligações leves.

Segundo (Freeman, 2010) o padrão observer, funciona como assinaturas de jornais e revistas:

- Uma editora começa a publicar jornais.
 - Você assina o jornal de uma editora e sempre recebe as novas edições. Enquanto você é um assinante, continua recebendo os jornais.
- Você cancela a assinatura quando não quer mais receber os jornais e eles param de ser entregues. Enquanto a editora continua no negócio, pessoas, hotéis, companhias aéreas e outras empresas fazem e cancelam assinaturas do jornal constantemente.

Vamos ver outro exemplo

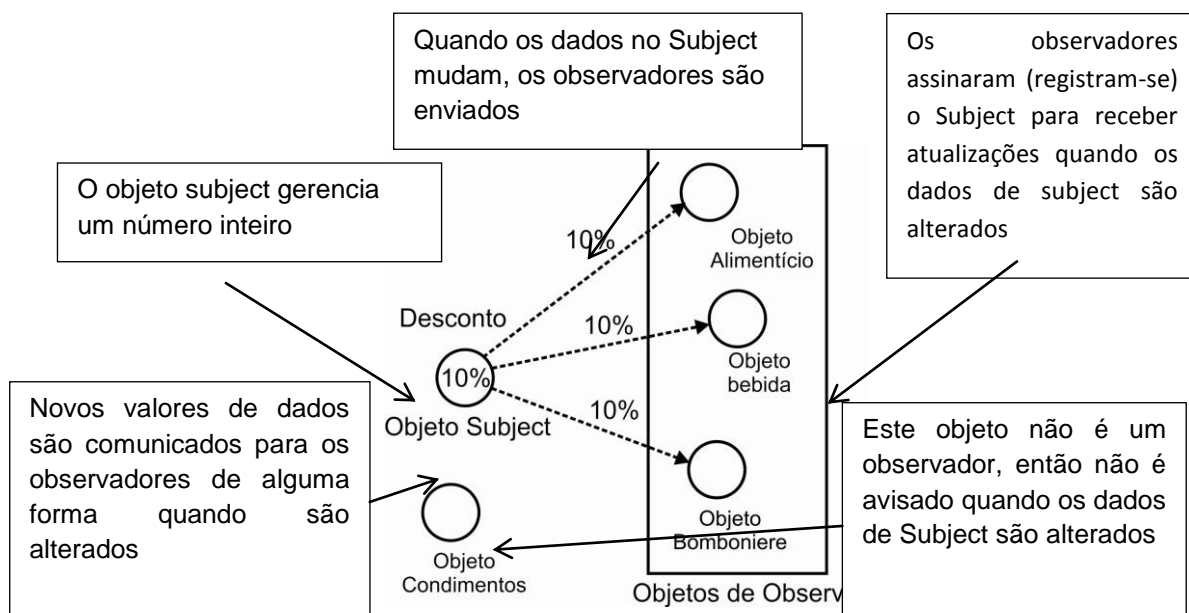


Figura 20 – O padrão Observer

Fonte: Adaptado (Freeman, 2010)

Vamos observar passo a passo como o padrão observer funciona (Freeman, 2010)

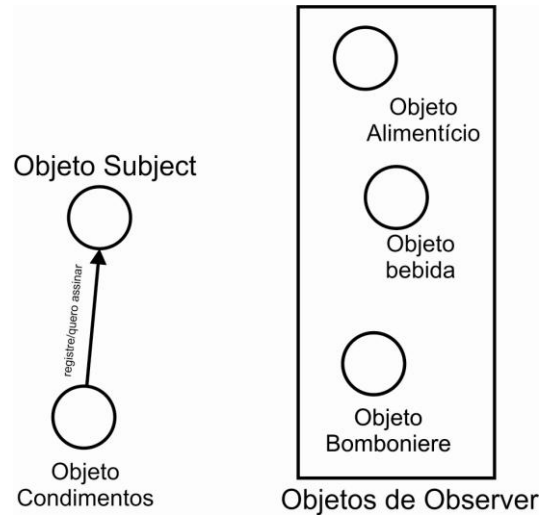


Figura 21 – Comunicação entre o objeto e o Subject

Fonte: Adaptado (Freeman, 2010)

O objeto condimento aparece e diz ao subject que deseja se tornar um observador.

O condimento agora é um observador oficial. Ele está na lista esperando pela próxima notificação de dados.

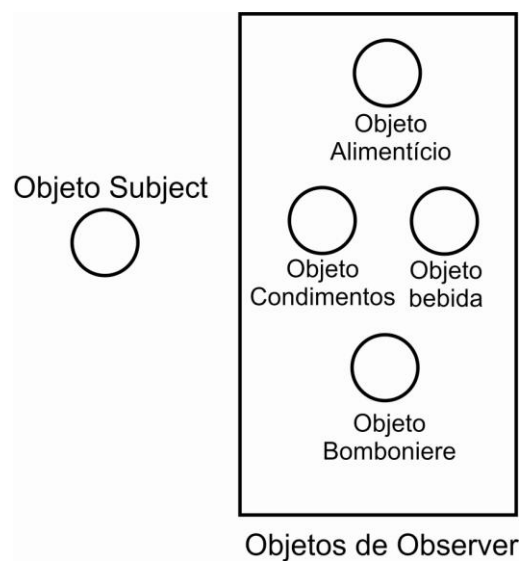


Figura 22 – Comunicação entre o objeto e o Subject

Fonte: Adaptado (Freeman, 2010)

O subject recebe um novo valor de dado e transmite para os seus observadores. Na figura 23 podemos observar um exemplo

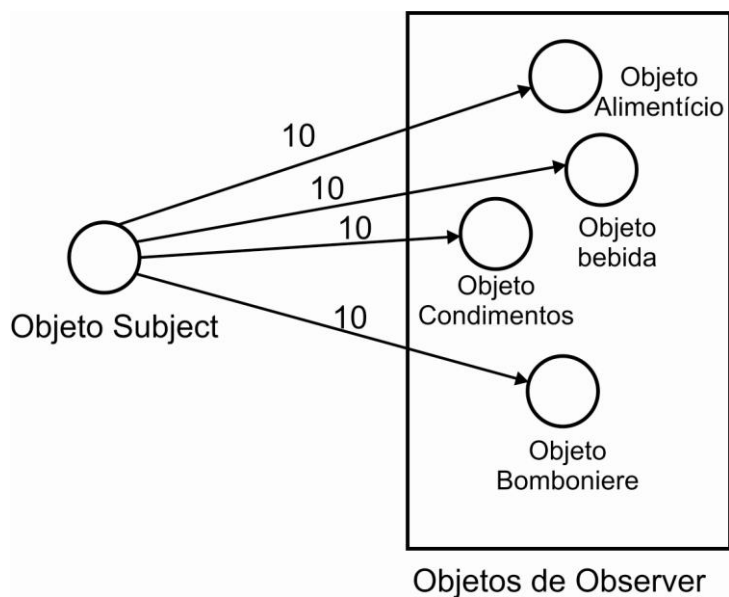


Figura 23 – Comunicação entre o objeto e o Subject
Fonte: Adaptado (Freeman, 2010)

O objeto Bomboniere pede para ser removido como observador. Este objeto recebe inteiros há algum tempo e não quer mais receber esses valores.

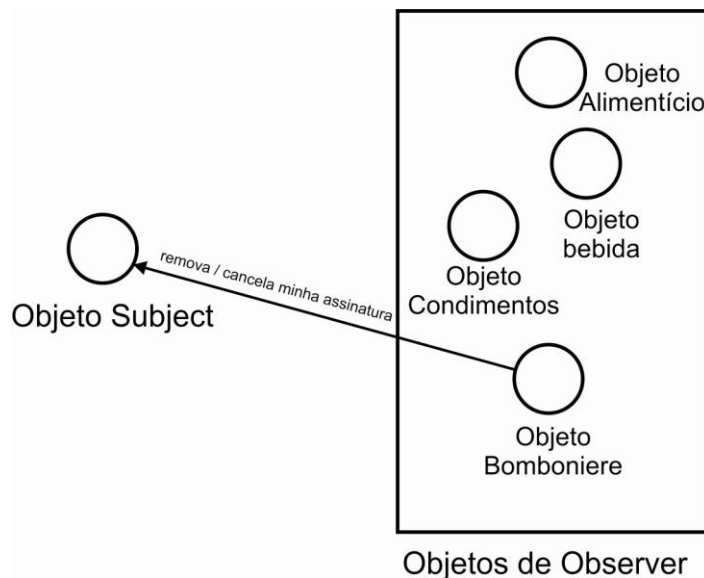


Figura 24 – Objeto Bomboniere pede para sair
Fonte: Adaptado (Freeman, 2010)

O objeto Bomboniere sai da lista dos observadores. Ele não receberá os próximos valores como mostra a Figura 25.

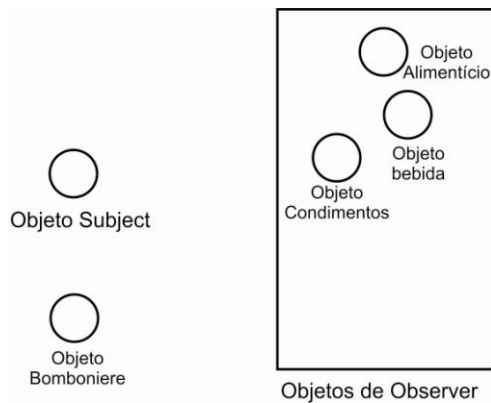


Figura 25 – Objeto Bomboniere sai
 Fonte: Adaptado (Freeman, 2010)

Definição do Padrão Observer: define a dependência um-para-muitos entre objetos para que quando um objeto mude de estado todos os seus dependentes sejam avisados e atualizados automaticamente.

Definindo o diagrama de classes do padrão observer.

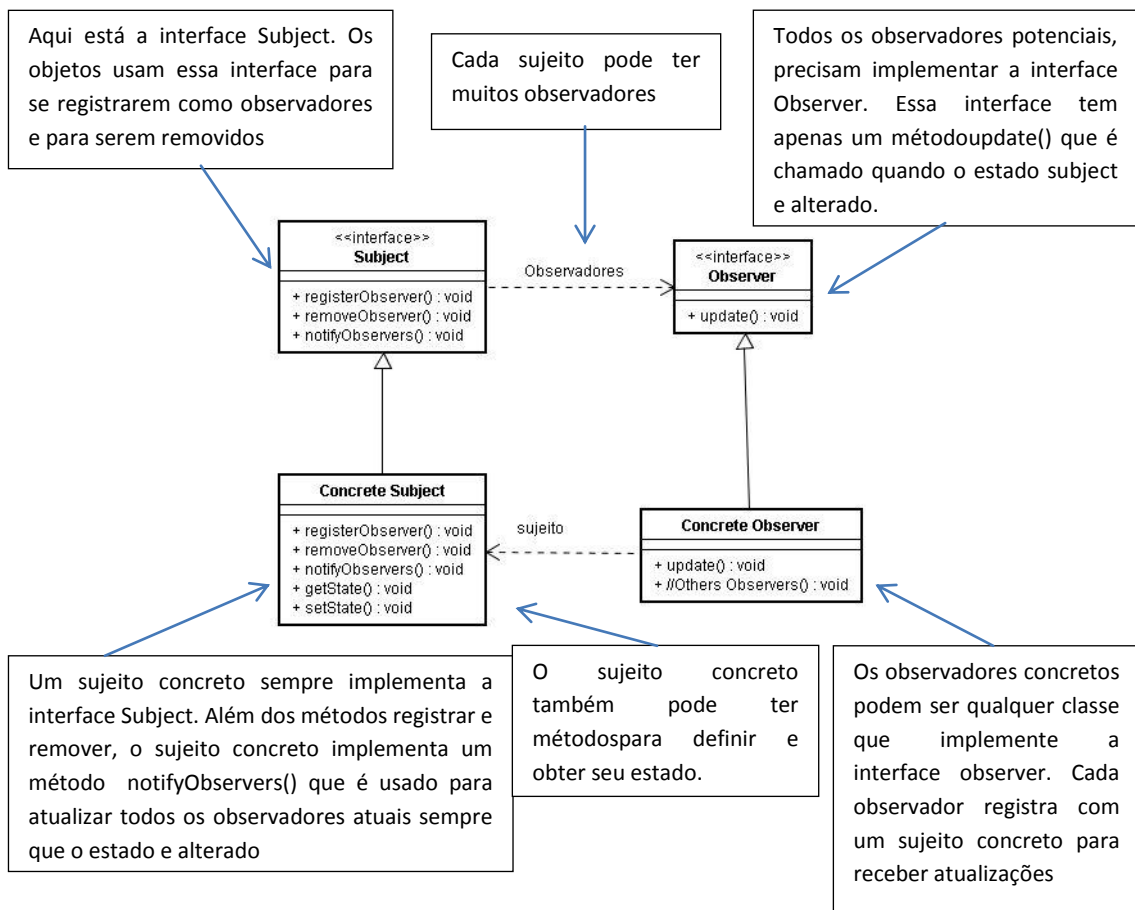


Figura 26 –Diagrama de classe do padrão Observer
Fonte: Adaptado (Freeman, 2010)

Você foi contratado para trabalhar em um sistema de software embarcado de um automóvel. Esse veículo possui dois sensores, um que realiza medidas de temperatura e umidade. Será necessário criar um objeto dados veículos que capture esses dados e passe para o display do veículo. A Figura 27 ilustra o contexto do problema.

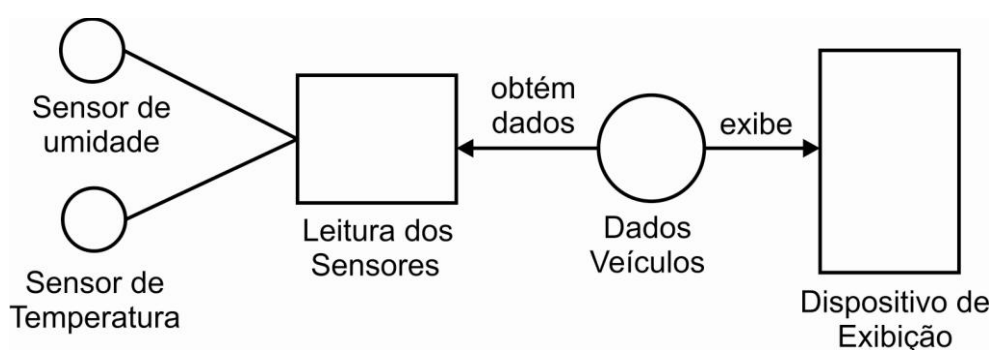


Figura 27 – Sistema de Leitura de Sensores
Fonte: Adaptado (Freeman, 2010)

Como mostrado na Figura 27, o objeto Leitura de Sensores, obtém os dados dos sensores de umidade e temperatura. O objeto Dados Veículos ele é o nosso subject que obtém os dados e envia para os observadores, que no nosso exemplo será o uma classe intitulada mostraMedidasAtuais e Mostraestatistica que estão dentro do dispositivo de exibição. Na Figura 28 apresenta-se o diagrama de classes.

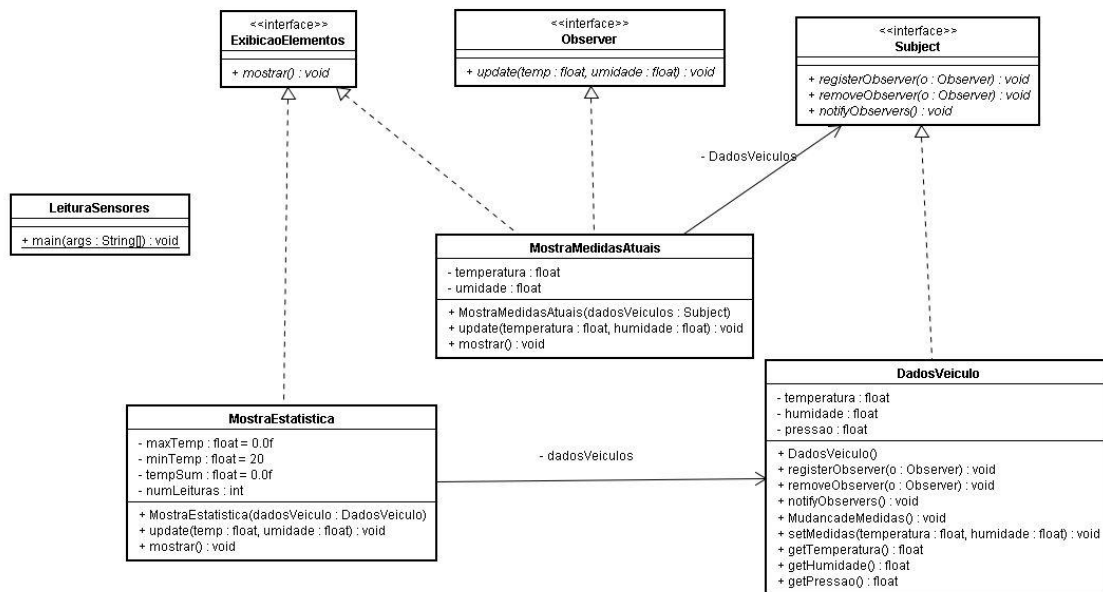


Figura 28 – Diagrama de Classe do Sistema de Leitura de Sensores.

Fonte: Adaptado (Freeman, 2010)

Na Figura 28 é apresentado três interfaces: ExibicaoElementos, Observer e Subject. Interface é um contrato que a classe que a implementa é obrigado a implementar seus métodos. No caso em tela, as classes Mostra Estatistica, implementa a interface ExibiçãoElementos e através da composição, passa os dados para a classe mostra veículos. A Listagem 10 mostra o código da interface.