RESPOSTAS DO EXERCICIO DE OBSERVER

1. O exemplo tem algo diferente?

A diferença entre os dois códigos é que o exemplo apresenta um método para notificar os observadores sobre alguma mudança de estado do comportamento do objeto.

1. Que outras formas poderíamos implementar o mesmo comportamento?

Poderíamos ter vários interessados em saber o comportamento do objeto, mas não serem notificados, sendo obrigados a sempre observarem o objeto, como também os interessados se registrarem para saber as alterações dos estados dos objetos.

3. Resumo do artigo de Martin Flower

Event Aggregator

O artigo trata de Canal de Eventos onde temos concentrados vários objetos em um único objeto a fim de simplificar o registro para clientes, ou seja dentro de um objeto teremos todos os eventos, e este fica sendo observado por vários objetos. É necessário que o cliente encontre o seu registro para cada objeto individualmente, se este tiver vários eventos será necessário registar cada um individualmente e uma assinatura separada.

O funcionamento do Event Aggregator é simples, você registra todos os objetos de origem que você está interessado tendo todos os objetos de destinos registrados nesse Event Aggregator. Ele converte, generaliza eventos que são espefícicos de um objeto de origem em um objeto genérico, garantindo aos observadores que não precisam se registrar em muitos tipos de eventos individuais.

Uma boa opção de uso é quando se tem muitos objetos, pois ao invés de ter observador lidando com o registro de todos eles, você apenas centraliza a lógica de para o Event Aggregator, fazendo com que se torne simples o registro e também os problemas de gerenciamento de memória e desempenho ao usar observadores.