



Design Patterns **(Padrões de Projeto)**

Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro
Prof. Edwar Saliba Júnior
Janeiro de 2020



Pergunta

- Qual é a única coisa com a qual podemos contar no desenvolvimento de software?
 - Não importa onde você trabalha, o que está criando ou em que linguagem está programando, qual é a constante que estará sempre com você?
 - **ALTERAÇÃO**



Manutenção

- Como foi visto no *slide* anterior, uma das poucas certezas que você terá sobre o sistema que desenvolver, é a de que ele será alterado em algum momento;
- Alterações podem ser feitas por vários motivos:
 - implementação de melhorias;
 - correção de inconsistências;
 - melhoria de performance e/ou utilização de recursos e
 - etc.

Padrões de Projeto

- Um Padrão de Projeto (PP) nomeia, abstrai e identifica os aspectos-chave de uma estrutura de projeto comum para torná-la útil para a criação de projeto orientado a objetos reutilizável;
- O PP identifica as classes e instâncias participantes, seus papéis, colaborações e distribuição de responsabilidades;
- Cada PP focaliza um problema ou tópico particular de projeto orientado a objetos;
- O PP descreve em que situação pode ser aplicado, se ele pode ser aplicado em função de outras restrições de projetos e as consequências, custos e benefícios de sua utilização.

Gamma et al. (2000, p. 20)



Padrão *Observer*

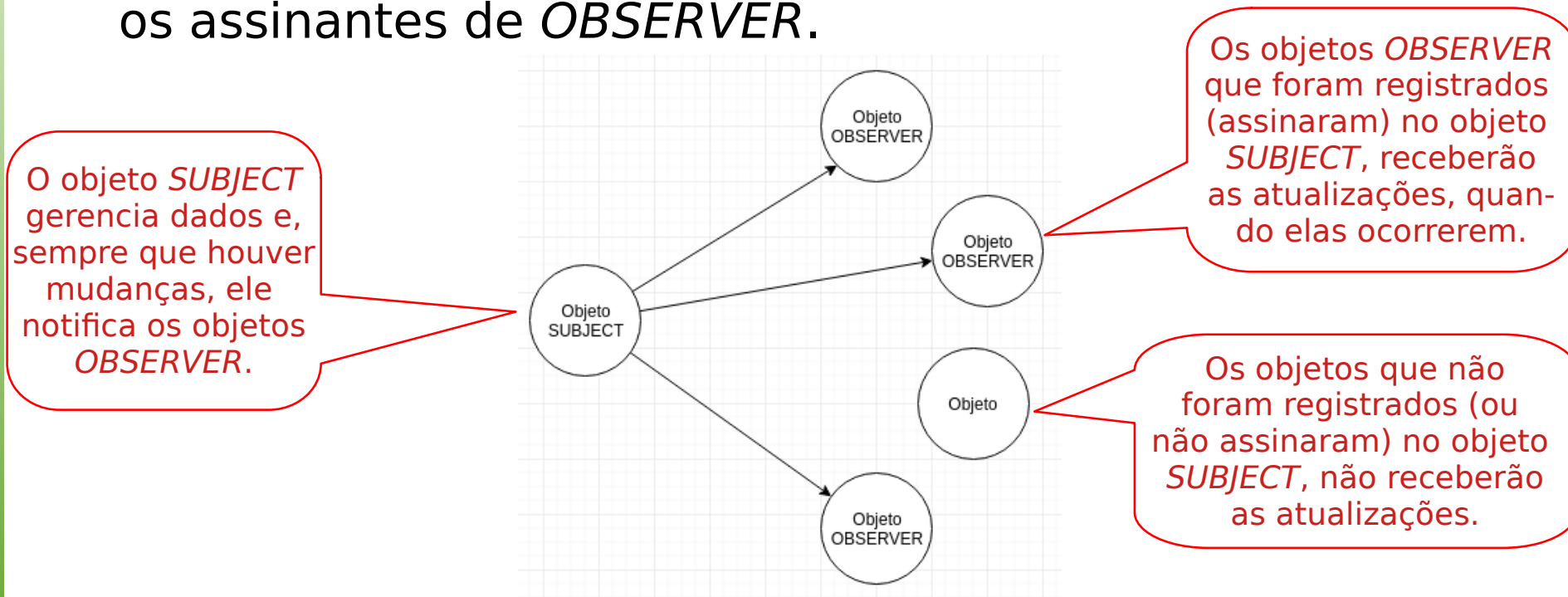
- **Contextualização:**

- Você sabe como as assinaturas de jornais e revistas funcionam?
- Funcionam assim:
 - uma editora começa a publicar jornais;
 - você assina o jornal de uma editora e sempre recebe as novas edições;
 - enquanto você for assinante, continuará recebendo as novas edições;
 - você cancela a assinatura e os jornais param de ser entregues;
 - enquanto a editora continua no negócio, entregando para hotéis, pessoas, companhias aéreas e etc.

Adaptado de Freeman e Freeman (2007, p. 56)

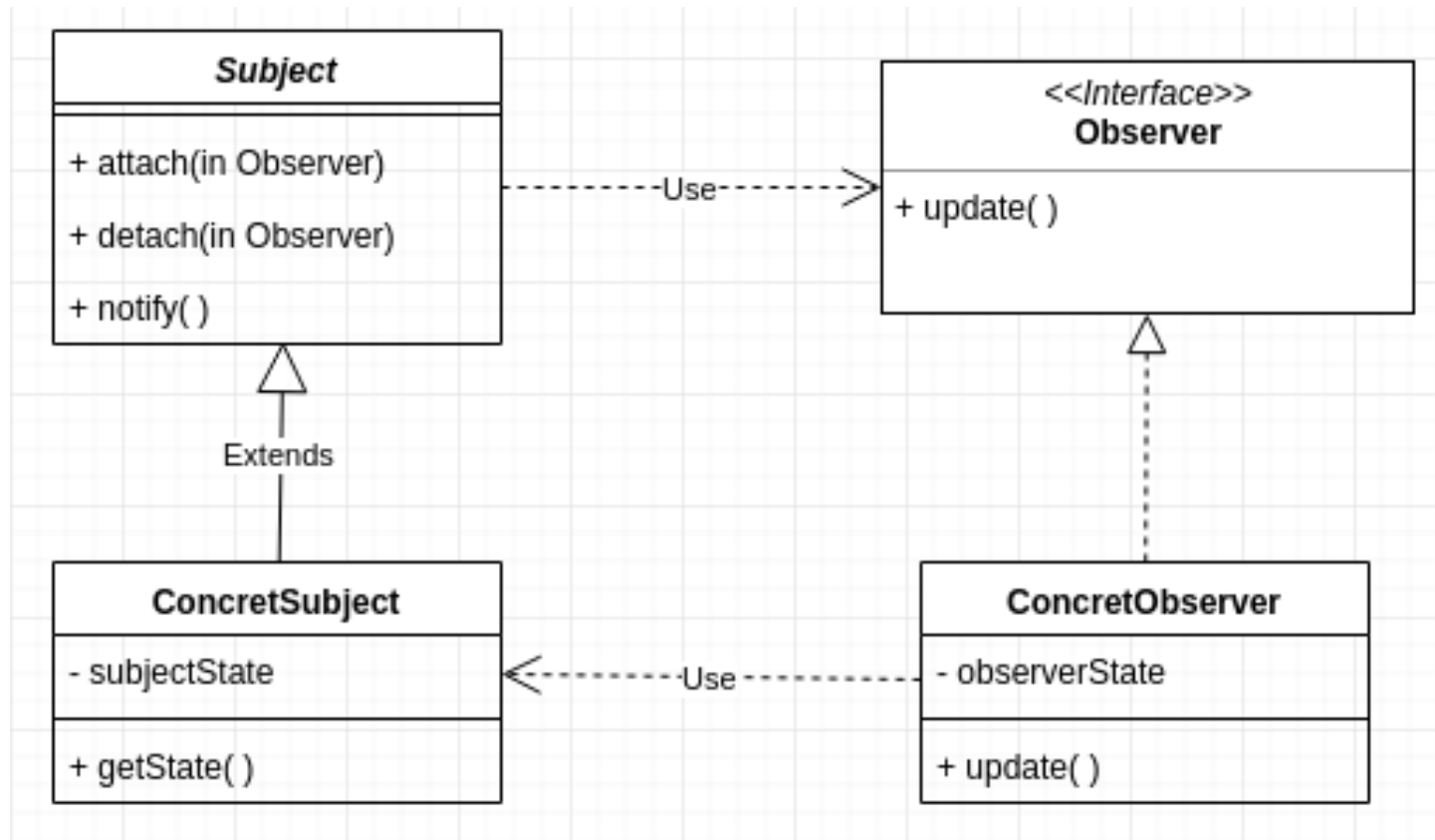
Padrão *Observer*

- Editora + assinantes = “padrão *observer*”
- Se você entende a assinatura de jornais, deve entender o padrão *observer*, onde chamamos a editora de *SUBJECT* e os assinantes de *OBSERVER*.



Adaptado de Freeman e Freeman (2007, p. 56)

Padrão *Observer*





Padrão *Observer*

- Exemplo utilizando recursos que funcionam até o Java 8:
 - [código-fonte](#)
- Exemplo utilizando recursos que funcionam do Java 9 em diante:
 - [código-fonte](#)

Outros Padrões

- *Design Patterns:*
 - 1) Decorator
 - 2) Factory
 - 3) Singleton
 - 4) Command 1 e 2
 - 5) Adapter
 - 6) Façade
 - 7) Template Method
 - 8) Iterador
 - 9) Composite
 - 10) State
 - 11) Proxy e
 - 12) Padrões Compostos.

Bibliografia

- COLLARES, Paulo. **Exemplo do Padrão de Projeto 'Comando': Controle Remoto**. Disponível em: <<https://www.paulocollares.com.br/programacao/exemplo-do-padrao-de-projeto-comando-controle-remoto/>>. Acesso em: 31 jan. 2020.
- FREEMAN, Eric; FREEMAN, Elisabeth. **Use a Cabeça! Padrões de Projetos (design Patterns)**. 2. ed. Rio de Janeiro: Alta Books, 2007.
- GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. **Padrões de Projeto: soluções reutilizáveis de software orientado a objetos**. Porto Alegre: Bookman, 2000.
- OPUS-SOFTWARE. **Singleton Design Pattern: o que é, onde usar e quais as vantagens**. Disponível em: <<https://www.opus-software.com.br/singleton-design-pattern/>>. Acesso em: 28 jan. 2020.
- UTLUIZ. **Qual a finalidade do Transient e Volatile no Java?**. Disponível em: <<https://pt.stackoverflow.com/questions/116047/qual-a-finalidade-do-transient-e-volatile-no-java>>. Acesso em: 28 jan. 2020.
- WIKIPÉDIA. **Command**. Disponível em: <<https://pt.wikipedia.org/wiki/Command>>. Acesso em: 29 jan. 2020.