

# Padrão Observer

Matheus Vianna Nunes

Gabriel De Albuquerque Basileu

# Definição

É um padrão que define uma cadeia de dependência de um pra muitos de forma em que quando ocorre mudança com o objeto observado, que é chamado de “Subject”, os objetos dependentes são notificados, estes são chamados de “Observers”.

O “Subject” notifica seus dependentes normalmente chamando um de seus métodos.

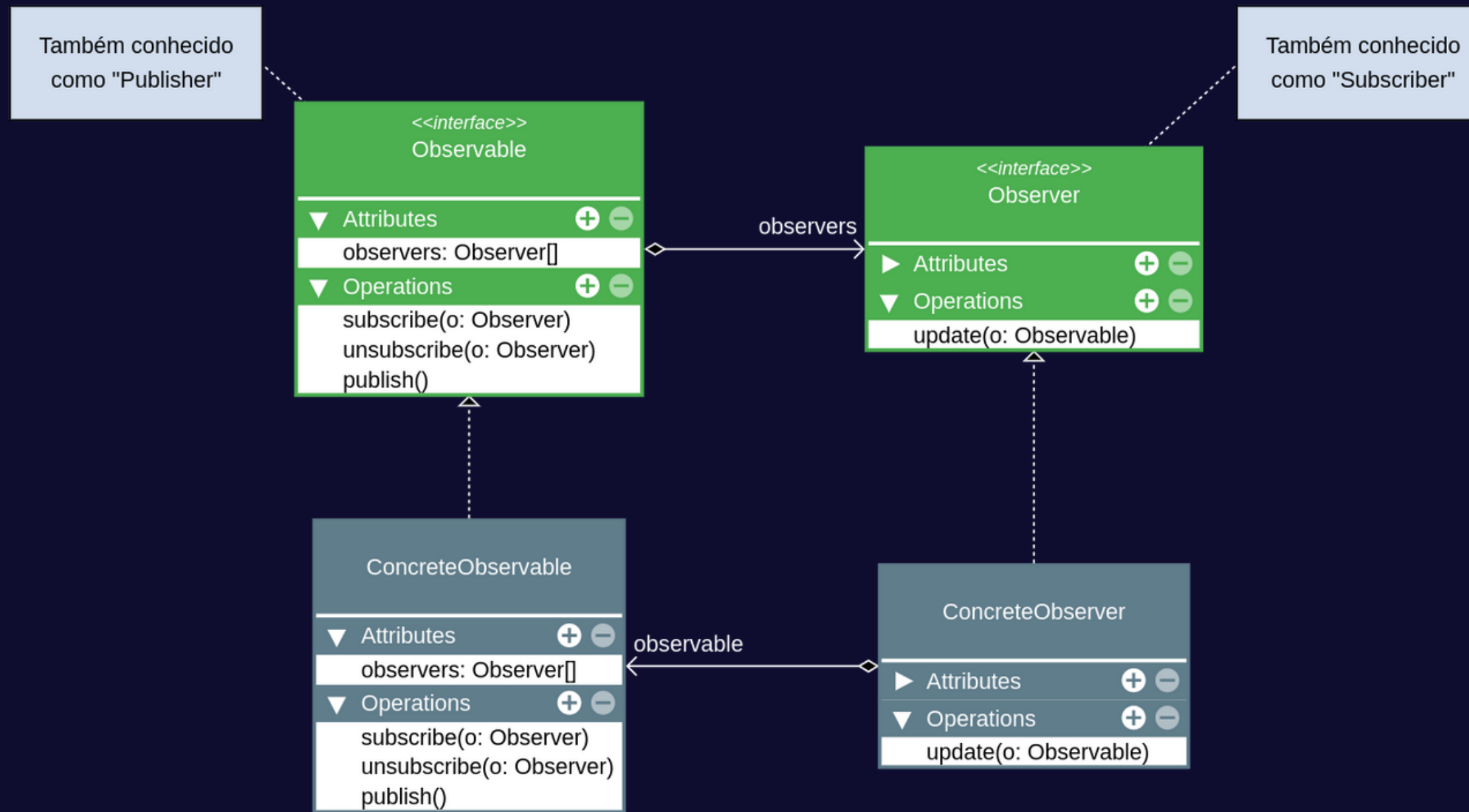
# Subject

- Subject é o objeto observado, por padrão é responsável por manter uma lista de seus dependentes e notifica-los eventualmente sobre suas mudanças de estado.

# Observer

- Os Observers são os objetos que recebem atualizações do Subject, podendo assim registrar-se no Subject para receber suas atualizações ou cancelar o registro assim deixando de receber atualizações.

# Modelagem Conceitual



# Relações com Outros Padrões

O Chain of Responsibility, Command, Mediator e Observer abrangem várias maneiras de se conectar remetentes e destinatários de pedidos:

- O Chain of Responsibility passa um pedido sequencialmente ao longo de um corrente dinâmica de potenciais destinatários até que um deles atua no pedido.
- O Command estabelece conexões unidirecionais entre remetentes e destinatários.
- O Mediator elimina as conexões diretas entre remetentes e destinatários, forçando-os a se comunicar indiretamente através de um objeto mediador.
- O Observer permite que destinatários inscrevam-se ou cancelem sua inscrição dinamicamente para receber pedidos.

# Exemplo Observer

```
1 package br.edu.femass;  
2  
3 public interface Subscriber {  
4     public void update(String s, String a);  
5 }
```

# Exemplo Subject

```
1 package br.edu.femass;  
2  
3 public interface Subject {  
4     public void registerObserver(Subscriber observer);  
5     public void unregisterObserver(Subscriber subscriber);  
6     public void notifyAllObserver(String notify);  
7 }
```

```

1  package br.edu.femass;
2  import java.util.ArrayList;
3
4  public class Account implements Subject {
5      private String accountName;
6      private ArrayList<Subscriber> followers;
7      public Account(String accountName) {
8          this.accountName = accountName;
9          followers = new ArrayList<Subscriber>();
10     }
11     @Override
12     public void registerObserver(Subscriber subscriber) {
13         if(subscriber != null){
14             this.followers.add(subscriber);
15         }
16         System.out.println(subscriber + " Started following " + accountName);
17     }
18
19     @Override
20     public void unregisterObserver(Subscriber subscriber) {
21         if(subscriber != null){
22             this.followers.remove(subscriber);
23         }
24         System.out.println(subscriber + " Stopped following " + accountName);
25     }
26
27     @Override
28     public void notifyAllObserver(String tweet) {
29         for(Subscriber follower : followers)
30             follower.update(accountName, tweet);
31     }
32     public void tweet(String tweet) {
33         System.out.println(accountName + " has tweeted: " + tweet + "\n");
34         notifyAllObserver(tweet);
35     }
36 }

```

# Exemplo Classe Concreta do Subject



# Exemplo Classe Concreta do Observer

```
1  package br.edu.femass;
2
3  ...
4  public class Follower implements Subscriber {
5      private String followerName;
6
7      public Follower(String followerName){
8          this.followerName = followerName;
9      }
10
11     @Override
12     public void update(String accountName, String tweet) {
13         System.out.println(followerName + " has received " + accountName + "'s tweet: " + tweet);
14     }
15
16     @java.lang.Override
17     public java.lang.String toString() {
18         return "Follower{" + followerName + "}";
19     }
20 }
```

# Exemplo de uso classe Main

```
1  package br.edu.femass;
2
3  ...
4  public class Main {
5      Run | Debug
6      public static void main(String[] args) {
7          //Criando as contas
8          Account matheus = new Account(accountName: "Matheus");
9          Account gabriel = new Account(accountName: "Gabriel");
10
11         //Criando os seguidores de Matheus
12         Follower Gustin = new Follower(followerName: "@Gustin");
13         Follower Mefiu = new Follower(followerName: "@Baby");
14         Follower Tozin = new Follower(followerName: "@Tozin");
15
16         //Criando os seguidores de Gabriel
17         Follower Baby = new Follower(followerName: "@Mefiu");
18         Follower Adryan = new Follower(followerName: "@Adryan");
19         Follower Dudu = new Follower(followerName: "@Pedro");
20
21         //Adicionando os seguidores a lista de 'seguidos' de Matheus
22         matheus.registerObserver(Gustin);
23         matheus.registerObserver(Mefiu);
24         matheus.registerObserver(Tozin);
25
26         //Adicionando os seguidores a lista de 'seguidos' de Gabriel
27         gabriel.registerObserver(Baby);
28         gabriel.registerObserver(Adryan);
29         gabriel.registerObserver(Dudu);
30
31         //Gerando um publicação no Twitter
32         matheus.tweet(tweet: "guys dont play league of legends, once you start you wont be able to do anything else... help");
33         gabriel.tweet(tweet: "Twitter is purging a lot of spam/scam accounts right now, so you may see your follower count drop");
34
35         //Retirando um seguidor da lista de 'seguidos' de Matheus
36         matheus.unregisterObserver(Gustin);
37     }
38 }
```

# Saída no Terminal

```
Follower{@Gustin} Started following Matheus
Follower{@Baby} Started following Matheus
Follower{@Tozin} Started following Matheus
Follower{@Mefiu} Started following Gabriel
Follower{@Adryan} Started following Gabriel
Follower{@Pedro} Started following Gabriel
Matheus has tweeted: guys dont play league of legends, once you start you wont be able to do anything else... help

@Gustin has received Matheus's tweet: guys dont play league of legends, once you start you wont be able to do anything else... help
@Baby has received Matheus's tweet: guys dont play league of legends, once you start you wont be able to do anything else... help
@Tozin has received Matheus's tweet: guys dont play league of legends, once you start you wont be able to do anything else... help
Gabriel has tweeted: Twitter is purging a lot of spam/scam accounts right now, so you may see your follower count drop

@Mefiu has received Gabriel's tweet: Twitter is purging a lot of spam/scam accounts right now, so you may see your follower count drop
@Adryan has received Gabriel's tweet: Twitter is purging a lot of spam/scam accounts right now, so you may see your follower count drop
@Pedro has received Gabriel's tweet: Twitter is purging a lot of spam/scam accounts right now, so you may see your follower count drop
Follower{@Gustin} Stopped following Matheus

Process finished with exit code 0
```

# Bibliografia

- Design Patterns: Elements of Reusable Object-Oriented Software
- <https://www.devmedia.com.br/padrao-de-projeto-observer-em-java/26163>
- <https://refactoring.guru/pt-br/design-patterns/observer>



## Repositórios:

<https://github.com/GabrielAL4/observer-pattern>

<https://github.com/Tue20022/observer-pattern>