Instituto Federal Catarinense *(Campus Blumenau)*
Professor: Ricardo de La Rocha Ladeira
Matéria: Padrões de Projeto
Nomes: Gabrielli Danker
Turma: BCC 2025.1
Data de entrega: 20 de Março de 2025

Exercícios

1) Acrescente um observador ao sistema de notificações climáticas.

   RESPOSTA: Foi adicionado um observador chamado Painel Digital de Rua, como no código abaixo:

```java
// Novo Observador - Painel Digital de Rua
class StreetPanel implements WeatherObserver {
    @Override
    public void update(String weatherUpdate) {
        System.out.println("[Painel Digital de Rua] Exibindo no painel: " + weatherUpdate);
    }
}

public class ObserverPatternExample {
    public static void main(String[] a) {
        WeatherService weatherService = new WeatherService();
        PhoneWidget phone = new PhoneWidget("Ricardo");
        NotebookWidget notebook = new NotebookWidget();
        SmartTV tv = new SmartTV();
        AlexaAssistant alexa = new AlexaAssistant();
        StreetPanel painel = new StreetPanel(); // Novo observador

        weatherService.addObserver(phone);
        weatherService.addObserver(notebook);
        weatherService.addObserver(tv);
        weatherService.addObserver(alexa);
        weatherService.addObserver(painel); // Adicionando o novo observador

        weatherService.setWeatherCondition("Chuva forte");
        weatherService.setWeatherCondition("Ensolarado com ventos fracos");
    }
```

   Que retornou isso:

```
[WeatherService] Nova condição do tempo: Chuva forte
[Celular - Ricardo] Notificação: O clima mudou para: Chuva forte
[Notebook Widget] Atualizando tela: Chuva forte
[Smart TV] Exibindo notificação: Chuva forte
[Alexa] Dizendo: 'Atenção, o clima mudou para Chuva forte.'
[Painel Digital de Rua] Exibindo no painel: Chuva forte

[WeatherService] Nova condição do tempo: Ensolarado com ventos fracos
[Celular - Ricardo] Notificação: O clima mudou para: Ensolarado com ventos fracos
[Notebook Widget] Atualizando tela: Ensolarado com ventos fracos
[Smart TV] Exibindo notificação: Ensolarado com ventos fracos
[Alexa] Dizendo: 'Atenção, o clima mudou para Ensolarado com ventos fracos.'
[Painel Digital de Rua] Exibindo no painel: Ensolarado com ventos fracos
```

   O código editado completo, está na pasta Observer -> ObserverPatternExample.java

2) Crie livremente algum outro exemplo que utilize o padrão de projeto *Observer*. A escolha da linguagem de programação é livre.

RESPOSTA: O tema escolhido foi de estoque de uma loja. A seguir tem o print do código:

```java
import java.util.ArrayList;
import java.util.List;

interface StockObserver {
    void update(String product, int quantity);
}

interface StockSubject {
    void addObserver(StockObserver observer);
    void removeObserver(StockObserver observer);
    void notifyObservers();
}

class Product implements StockSubject {
    private List<StockObserver> observers = new ArrayList<>();
    private String name;
    private int quantity;

    public Product(String name, int quantity) {
        this.name = name;
        this.quantity = quantity;
    }

    public void addObserver(StockObserver observer) {
        observers.add(observer);
    }

    public void removeObserver(StockObserver observer) {
        observers.remove(observer);
    }

    public void notifyObservers() {
        for (StockObserver observer : observers) {
            observer.update(name, quantity);
        }
    }

    public void sellItem() {
        if (quantity > 0) {
            quantity--;
            System.out.println("\n[Loja] Produto vendido: " + name + " | Estoque restante: " + quantity);
            if (quantity <= 3) { // Quando o estoque chega a 3 ou menos, avisa os interessados
                notifyObservers();
            }
        } else {
            System.out.println("\n[Loja] Produto esgotado: " + name);
        }
    }
}

// Observador - Cliente
class Customer implements StockObserver {
    private String name;

    public Customer(String name) {
        this.name = name;
    }

```

```java
    @Override
    public void update(String product, int quantity) {
        System.out.println("[Cliente " + name + "] Alerta! O estoque de " + product + " está baixo (" + quantity + " restantes). Compre logo!");
    }
}

// Observador - Gerente
class Manager implements StockObserver {
    @Override
    public void update(String product, int quantity) {
        System.out.println("[Gerente] Atenção! O estoque de " + product + " está baixo. Avaliar necessidade de reposição.");
    }
}

// Observador - Sistema de Reposição
class RestockSystem implements StockObserver {
    @Override
    public void update(String product, int quantity) {
        System.out.println("[Sistema de Reposição] Ordem de compra gerada para o produto: " + product);
    }
}

public class StockObserverExample {
    public static void main(String[] args) {
        Product laptop = new Product("Laptop Gamer", 5);

        Customer cliente1 = new Customer("Ricardo");
        Customer cliente2 = new Customer("Ana");
        Manager gerente = new Manager();
        RestockSystem sistemaReposicao = new RestockSystem();

        laptop.addObserver(cliente1);
        laptop.addObserver(cliente2);
        laptop.addObserver(gerente);
        laptop.addObserver(sistemaReposicao);

        // Simulando vendas
        laptop.sellItem();
        laptop.sellItem();
        laptop.sellItem();
        laptop.sellItem();
    }
}
```

Este código retorna isso:

```
[Loja] Produto vendido: Laptop Gamer | Estoque restante: 4

[Loja] Produto vendido: Laptop Gamer | Estoque restante: 3
[Cliente Ricardo] Alerta! O estoque de Laptop Gamer está baixo (3 restantes). Compre logo!
[Cliente Ana] Alerta! O estoque de Laptop Gamer está baixo (3 restantes). Compre logo!
[Gerente] Atenção! O estoque de Laptop Gamer está baixo. Avaliar necessidade de reposição.
[Sistema de Reposição] Ordem de compra gerada para o produto: Laptop Gamer

[Loja] Produto vendido: Laptop Gamer | Estoque restante: 2
[Cliente Ricardo] Alerta! O estoque de Laptop Gamer está baixo (2 restantes). Compre logo!
[Cliente Ana] Alerta! O estoque de Laptop Gamer está baixo (2 restantes). Compre logo!
[Gerente] Atenção! O estoque de Laptop Gamer está baixo. Avaliar necessidade de reposição.
[Sistema de Reposição] Ordem de compra gerada para o produto: Laptop Gamer

[Loja] Produto vendido: Laptop Gamer | Estoque restante: 1
[Cliente Ricardo] Alerta! O estoque de Laptop Gamer está baixo (1 restantes). Compre logo!
[Cliente Ana] Alerta! O estoque de Laptop Gamer está baixo (1 restantes). Compre logo!
[Gerente] Atenção! O estoque de Laptop Gamer está baixo. Avaliar necessidade de reposição.
[Sistema de Reposição] Ordem de compra gerada para o produto: Laptop Gamer
```

O código está na pasta Observer -> StockObserverExample.java.