



UNIVERSIDADE FEDERAL DO CEARÁ

Universidade Federal do Ceará - *campus* Quixadá

Reuso de Software

Artigo de resposta para a Atividade 03, usando o padrão observer

Docente:

JEFFERSON DE CARVALHO SILVA

Discentes:

Francisco Robson Queiroz Mendes

Sistema de Alerta para Saúde em Wearables Utilizando o Padrão de Projeto Observer

Resumo

Wearables, como smartwatches e pulseiras inteligentes, têm se tornado essenciais para o monitoramento de saúde em tempo real. Este artigo aborda como o padrão de projeto **Observer** pode ser utilizado para implementar um sistema de alerta de saúde em wearables, promovendo flexibilidade e desacoplamento entre sensores (sujeito) e os componentes que reagem às mudanças (observadores). Apresentamos um exemplo prático em Java para ilustrar o conceito.

Introdução

Os wearables são dispositivos inteligentes que monitoram sinais vitais, como frequência cardíaca, oxigenação do sangue e níveis de estresse. Uma característica fundamental desses sistemas é a capacidade de notificar o usuário ou serviços médicos em casos críticos, como um aumento repentino na frequência cardíaca.

O desafio é criar um sistema que não dependa diretamente das classes específicas de sensores ou notificadores. Por exemplo, o wearable pode inicialmente alertar apenas o usuário, mas no futuro pode ser integrado a serviços médicos ou sistemas de emergência. O padrão **Observer** é uma solução ideal para este problema, pois promove o desacoplamento e a extensibilidade.

Contexto e Problema

O sistema monitora os sinais vitais de um usuário e deve alertar:

- **Usuário:** Em casos de alterações leves ou moderadas.

-
- **Serviços médicos:** Em casos de emergência, como frequência cardíaca muito alta.

Problema: O wearable não deve ser acoplado a um conjunto fixo de notificadoros, pois diferentes cenários podem exigir novas implementações. Além disso, o sistema deve permitir adicionar novos notificadoros sem alterar o código do wearable, promovendo flexibilidade e manutenibilidade.

Solução: O padrão Observer separa o wearable (sujeito) dos sistemas que consomem suas notificações (observadores). Quando o wearable detecta uma alteração, ele notifica automaticamente todos os observadores registrados.

O Padrão Observer

O padrão **Observer** estabelece uma relação do tipo um-para-muitos entre objetos. Quando o estado de um objeto (sujeito) muda, todos os objetos dependentes (observadores) são automaticamente notificados.

Elementos principais:

- **Sujeito (Subject):** Armazena uma lista de observadores e notifica-os sobre mudanças.
- **Observadores (Observers):** Reagem às notificações do sujeito.
- **Interface Observer:** Define o contrato que todos os observadores devem implementar.

No caso deste sistema, o wearable é o sujeito, e os observadores incluem classes como alertas para o usuário ou notificações para serviços médicos.

Implementação do exercício em Java

A seguir como o padrão Observer pode ser aplicado em Java para este contexto medico.

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



```
// Interface Observer
interface Observer {
    void update(Wearable subject);
}

// Classe Sujeito (Subject)
class Wearable {
    private List<Observer> observers = new ArrayList<>();
    private int heartRate;

    public void attach(Observer observer) {
        observers.add(observer);
    }

    public void detach(Observer observer) {
        observers.remove(observer);
    }

    public void notifyObservers() {
        for (Observer observer : observers) {
            observer.update(this);
        }
    }

    public int getHeartRate() {
        return heartRate;
    }

    public void setHeartRate(int heartRate) {
        this.heartRate = heartRate;
        notifyObservers();
    }
}

// Classe Observador: Alerta para o Usuário
class UserAlert implements Observer {
    @Override
    public void update(Wearable subject) {
        System.out.println("User Alert: Heart rate is " +
subject.getHeartRate() + " bpm!");
    }
}
```



```
// Classe Observador: Serviço Médico
class MedicalService implements Observer {
    @Override
    public void update(Wearable subject) {
        if (subject.getHeartRate() > 100) {
            System.out.println("Medical Service Alert: Critical
heart rate detected: " + subject.getHeartRate() + " bpm!");
        }
    }
}

// Classe Principal (Main)
public class HealthAlertSystem {
    public static void main(String[] args) {
        Wearable wearable = new Wearable();

        Observer userAlert = new UserAlert();
        Observer medicalService = new MedicalService();

        wearable.attach(userAlert);
        wearable.attach(medicalService);

        System.out.println("Heart rate set to 90 bpm.");
        wearable.setHeartRate(90);

        System.out.println("Heart rate set to 110 bpm.");
        wearable.setHeartRate(110);
    }
}
```

Resultado

A execução do sistema gera o seguinte resultado:

```
Heart rate set to 90 bpm.
User Alert: Heart rate is 90 bpm!
Heart rate set to 110 bpm.
User Alert: Heart rate is 110 bpm!
Medical Service Alert: Critical heart rate detected: 110 bpm!
```

Benefícios do Padrão Observer

1. **Desacoplamento:** O wearable não depende diretamente de classes específicas de notificadores.
2. **Extensibilidade:** É fácil adicionar novos observadores, como sistemas de análise de dados ou notificações para familiares, sem modificar o wearable.
3. **Reutilização:** O mesmo mecanismo pode ser usado para outros sinais vitais, como oxigenação ou temperatura corporal.

Conclusão

O padrão Observer é uma solução para sistemas que necessitam de reações dinâmicas a mudanças de estado. No contexto de wearables, ele garante flexibilidade, manutenibilidade e facilita a adição de novos comportamentos ao sistema. Esta abordagem é escalável e pode ser estendida para monitorar diversos sinais vitais, oferecendo maior segurança e personalização aos usuários.