

06

Implementando Chain of Responsibility em Requisições Web

Um servidor de aplicação bancária que se comunica com outros deve responder de várias formas diferentes, de acordo com a solicitação da aplicação cliente.

Se a aplicação solicitar uma `Conta`, cujos atributos são separados por ponto-e-vírgula, por exemplo, o servidor deverá serializar a conta nesse formato; se a aplicação solicitar XML, o servidor deverá serializar em XML; se ela pedir separado por % (por cento), a aplicação deverá devolver dessa forma.

Por exemplo: a versão em XML deve imprimir na tela: `<conta><titular>João da Silva</titular><saldo>1500.0</saldo></conta>`.

Implemente um `Chain of Responsibility` onde, dada uma requisição e uma conta bancária, ela passeia por toda a corrente até encontrar a classe que deve processar a requisição de acordo com o `formato` solicitado, e imprime na tela a conta bancária no formato correto.

Imagine que a classe `Requisição` possui um getter `getFormato()`, que responde "XML", "CSV", ou "PORCENTO", indicando qual tratamento adequado. Uma `Conta` possui apenas saldo e nome do titular:

```
enum Formato {
    XML,
    CSV,
    PORCENTO
}

class Requisicao {
    private Formato formato;
    public Requisicao(Formato formato) {
        this.formato = formato;
    }
    // getter para o Formato
}
```

A sua interface do `Chain of Responsibility` deve ser algo como:

```
interface Resposta {
    void responde(Requisicao req, Conta conta);
    void setProxima(Resposta resposta);
}
```