

Problema do cálculo do valor de um estacionamento

Considere o sistema de um estacionamento que precisa utilizar diversos critérios: Para calcular o valor que deve ser cobrado de seus clientes. Para um veículo de passeio, o valor deve ser calculado como R\$2,00 por hora, porém, caso o tempo seja maior do que 12 horas, será cobrada uma taxa diária, e caso o número de dias for maior que 15 dias, será cobrada uma mensalidade. Existem também regras diferentes para caminhões, que dependem do número de eixos e do valor da carga carregada, e para veículos para muitos passageiros, como ônibus e vans. O código a seguir apresenta um exemplo de como isto estava desenvolvido.

```
package br.com.fatec;

public class ContaEstacionamento {

    private Veiculo veiculo;
    private long inicio, fim;
    private final double HORA = 3600000;
    private final double DIA = 86400000;
    private final double MES = 2592E6;

    public double valorConta() {
        double value = 0;
        long atual = (fim == 0) ? System.currentTimeMillis() : fim;
        long periodo = inicio - atual;
        if (veiculo instanceof Passeio) {
            if (periodo < 12 * HORA) {
                value = 2.0 * Math.ceil(periodo / HORA);
            } else if (periodo > 12 * HORA && periodo < 15 * DIA) {
                value = 26.0 * Math.ceil(periodo / DIA);
            } else {
                value = 300.0 * Math.ceil(periodo / MES);
            }
        } else if (veiculo instanceof Carga) {
            // outras regras para veículos de carga
        }
        return value;
    }
}
```

Como é possível observar, o código utilizado para calcular os diversos condicionais é complicado de se entender. Apesar de apenas parte da implementação do método ter sido apresentada, pode-se perceber que a solução utilizada faz com que o método valorConta() seja bem

grande. As instâncias da classe ContaEstacionamento relacionadas com os veículos que estão no momento estacionados são exibidas para o operador e têm seu tempo atualizado periodicamente.

Até então, o próprio desenvolvedor sabia que o código estava ruim, mas como o código estava funcionando, preferiu deixar da forma que está. Porém, o software começou a ser vendido para outras empresas e outras regras precisariam ser incluídas.

Alguns municípios possuem leis específicas a respeito do intervalo de tempo para o qual um estacionamento deve definir sua tarifa. Além disso, diferentes empresas podem possuir diferentes critérios para cobrar o serviço de seus clientes.

A solução da forma como está não irá escalar para um número grande de regras! O código que já não estava bom poderia crescer de uma forma descontrolada e se tornar não gerenciável. O desenvolvedor sabia que precisaria refazer esse código para uma solução diferente. O que ele poderia fazer?

- Elabore uma solução para modificar a estrutura do sistema apresentado de forma que ela se torne flexível e resolva os problemas supracitados.