Prova de programação Java

O objetivo é avaliar como você vai desenvolver o código em termos de estilo, eficiência e qualidade.

histórico

05 Ago 2015	0.1	Versão Inicial	Enrique Pereira

Sumário

estrutura de avaliação
- requisitos
- recomendações
- entregáveis
- avaliação

tarefa

requisitos

- Não é necessário produzir uma aplicação completamente funcional, apenas uma implementação do problema proposto.
- Não se preocupe com interfaces gráficas nem com persistência em banco de dados.
- O código deve ser produzido em Java.
- O código deve ser submetido preferencialmente utilizando uma ferramenta de build e gestor de dependências, ou as dependências devem estar acompanhando o projeto.
- A implementação deve atender aos casos de testes providos.
- Criação de testes unitários para validar os requisitos do problema
- A solução não pode conter arquivos de IDE.

recomendações

- O trabalho envolve tomar algumas decisões. Tenha em mente os trade-offs necessários e busque clareza.
- Descreva brevemente e com clareza as razões que o levaram a fazer certas escolhas
- Um design limpo - algo como facilmente entendível por programadores jr. - deve ser preferido em relação a soluções muito complexas com ligeiros ganhos em eficiência. Entretanto uma solução mal desenhada será penalizada se houver uma solução 'padrão' não utilizada.
- Evite comentários óbvios.
- Códigos estranhos ou complexos podem ter um comentário descrevendo a solução.
- É permitido o uso de frameworks, desde que eles não implementem a solução central do problema.
- Tenha em mente que evitar acoplamento e aumentar coesão são importantes.

entregáveis

- Pequena documentação explicando suas decisões arquiteturais e versões de linguagem e ferramentas utilizadas
- Crie um projeto no seu Github para que vejamos os passos feitos através dos commits para resolver a tarefa.

avaliação

O time da RF verificará se a solução contempla os casos informados e estão dentro dos padrões de qualidade de código esperados para o cargo que desempenhará na empresa. Serão analisadas a implementação e as decisões tomadas.

tarefa

O objetivo dessa tarefa é avaliar como você vai desenvolver o código em termos de estilo, eficiência e qualidade.

Crie um projeto no seu Github para que vejamos os passos feitos através dos commits para resolver a tarefa.

A tarefa é a seguinte:

Desenvolver um sistema de agendamento de transferências financeiras.

1) O usuário deve poder agendar uma transferência financeira com as seguintes informações:

- Conta de origem (padrão XXXXX-X)
- Conta de destino (padrão XXXXX-X)
- Valor da transferência
- Taxa (a ser calculada)
- Data do agendamento
- Tipo (A, B, C, D)

2) Cada tipo de transação segue uma regra diferente para cálculo da taxa

A: Operações do tipo A tem uma taxa de $2 mais 3% do valor da transferência

B: Operações do tipo B tem uma taxa de:
$10 para pedidos com agendamento até 30 dias da data de cadastro
$8 para os demais

C: Operações do tipo C tem uma taxa regressiva conforme a data de agendamento:
maior que 30 dias da data de cadastro - 1.2%
até 30 dias da data de cadastro - 2.1%
até 25 dias da data de cadastro - 4.3%
até 20 dias da data de cadastro - 5.4%
até 15 dias da data de cadastro - 6.7%
até 10 dias da data de cadastro - 7.4%
até 5 dias da data de cadastro - 8.3%

D: Operações do tipo D tem a taxa igual a A, B ou C dependendo do valor da transferência.
Valores até $25.000 seguem a taxação tipo A
Valores de $25.001 até $120.000 seguem a taxação tipo B
Valores maiores que $120.000 seguem a taxação tipo C

3) O usuário deve poder ver todos os agendamentos cadastrados.

Nota: A persistência não precisa ser em banco de dados.

Fique à vontade para criar em cima desses requisitos.

Boa sorte!