Projeto final da disciplina de JAVA 1

Regras para entrega

A versão final do trabalho, incluindo os diagramas UML, deverão ser entregues através de um **link para o repositório no github** até o dia 02/04 (último dia de aula da disciplina). Nesse dia, pelo menos uma pessoa do grupo deverá apresentar o trabalho funcionando durante a aula, a apresentação deverá ter, no máximo, 10 minutos de duração.

O uso do Github é obrigatório, pois através dele conseguirei ver se todos os membros do grupo contribuíram para o projeto. **As pessoas que não tiverem contribuição, receberão nota 0 no projeto e, consequentemente, estarão reprovadas na disciplina.**

Caso precisem aprender sobre git e github, sugiro os seguintes vídeos (Github desktop):

Curso de Git e GitHub: grátis, prático e sem usar comandos no terminal - YouTube

Como usar o Github Desktop

GitHub + GitHub Desktop Aula 01 Criar repositório, clonar, commit e push

Obs.: Sugiro que desenvolvam em branches(ramificações) diferentes enquanto estiverem desenvolvendo diversas funcionalidades ao mesmo tempo e depois façam merge na branch principal quando terminarem o desenvolvimento daquela funcionalidade específica.

Sistema Bancário

O trabalho final da disciplina de Java 1 (orientação a objetos) consiste em uma atividade contínua paralela aos exercícios propostos durante as aulas.

O objetivo deste trabalho é promover o desenvolvimento de um pequeno sistema bancário utilizando os tópicos apresentados e desenvolvidos na disciplina.

Requisitos Mínimos - Java (52 pontos):

Classes

O sistema deverá ter as seguintes classes obrigatoriamente. (Podendo ter classes adicionais perante a necessidade descoberta por cada grupo) :

- Cliente
- Conta
- Conta Corrente
- Conta Poupança
- Funcionário

    - Gerente
    - Diretor
    - Presidente
- SistemaInterno (Classe main)

Atributos

- Cliente deve ter os atributos de senha e CPF para que possam logar no sistema interno.
- Conta, que deverá ser uma classe abstrata, deve conter os atributos de CPF do titular, para relacionar a conta com o usuário logado no sistema, e o saldo. Adicionalmente, a conta deve ter um atributo identificador da agência.
- Conta Corrente e Conta Poupança, que herdarão os atributos e métodos de Conta, devem conter um atributo "tipo" para identificação do tipo de conta.
- Funcionário (classe abstrata), deve conter atributos também de CPF e senha para que possam logar no sistema. Um atributo "cargo" também deve existir para identificar qual é o cargo daquele funcionário. Este atributo pode existir na própria classe Funcionario ou diretamente em suas classes filhas.
- Gerente, que estenderá de Funcionário deve ter um atributo de identificação da agência que é responsável por gerir.

Características de Funcionamento

Este sistema será executado como um menu interativo no console.

Na versão final (a ser entregue) é esperado que o sistema seja populado com os valores referentes aos clientes, contas e funcionários através da leitura de um arquivo de texto. Para testes iniciais, entretanto, valores podem ser atribuídos diretamente (através de um hashmap, por exemplo).

Obs.: O sistema não precisará realizar cadastros de clientes e/ou funcionários, somente login e as operações acima são suficientes.

Passos de Execução

No menu inicial o usuário deverá fornecer seu CPF e senha para logar.

O sistema deve ser capaz de identificar, no momento do login, se o usuário é um Cliente, Gerente, Diretor ou Presidente.

Caso 1:

Caso o usuário seja um Cliente, o sistema fornecerá um menu com opções de :

1. Movimentações na Conta

a. Saque

b. Depósito

c. Transferência para outra conta

2. Relatórios

a. Saldo. O sistema deverá imprimir o saldo na tela do terminal;

b. Relatório de tributação da conta corrente

i. O relatório de tributação deverá apresentar o total gasto nas operações até o momento do relatório.

ii. Adicionalmente deverão ser informados os valores que o banco cobra por cada operação bancária;

iii. Para cada saque será cobrado o valor de R$0.10 (dez centavos);

iv. Para cada depósito será cobrado o valor de R$0.10 (dez centavos);

v. Para cada transferência será cobrado o valor de R$0.20 (dez centavos) que deverá ser cobrado apenas do remetente;

c. Relatório de Rendimento da poupança

i. Retorna uma simulação do valor de rendimento da poupança no prazo informado. Neste caso o cliente deverá informar o valor em dinheiro e a

ii. quantidade de dias que pretende simular. O sistema deverá informar o

iii. rendimento desse dinheiro para o prazo informado;

d. Desafio:

i. Criar uma classe seguro de vida que possa ser contratado pelo cliente onde o mesmo informa o valor que será segurado. No ato da contratação será debitado 20% do valor contratado como tributo do seguro;

ii. Incluir no relatório de tributação o valor referente ao seguro de vida, caso este cliente possua estas informações.

Caso 2:

Caso o usuário seja um Funcionário no cargo de **Gerente** o sistema fornecerá um menu com opções de :

1. Movimentações e Informações da Conta

a. Saque

b. Depósito

c. Transferência para outra conta

2. Relatórios

    a. Saldo. O sistema deverá imprimir o saldo na tela do terminal;

    b. Relatório de tributação da conta corrente

        i. O relatório de tributação deverá apresentar o total gasto nas operações até o momento do relatório.

        ii. Adicionalmente deverão ser informados os valores que o banco cobra por cada operação bancária;

        iii. Para cada saque será cobrado o valor de R$0.10 (dez centavos);

        iv. Para cada depósito será cobrado o valor de R$0.10 (dez centavos);

        v. Para cada transferência será cobrado o valor de R$0.20 (dez centavos) que deverá ser cobrado apenas do remetente;

    c. Relatório de Rendimento da poupança

        i. Retorna uma simulação do valor de rendimento da poupança no prazo informado. Neste caso o cliente deverá informar o valor em dinheiro e a quantidade de dias que pretende simular. O sistema deverá informar o rendimento desse dinheiro para o prazo informado;

    d. Relatório do número contas na mesma agência em que este gerente trabalha

Caso 3:

Caso o usuário seja um Funcionário no cargo de **Diretor** o sistema fornecerá um menu com todas as opções anteriores, adicionando :

1. Relatórios

    a. Relatório com as informações de Nome, CPF e Agência de todos os clientes do sistema em ordem alfabética.

Caso 4:

Por fim, caso o usuário acessando o sistema seja um Funcionário no cargo de Presidente o sistema fornecerá um menu com todas as opções anteriores, adicionando :

1. Relatórios

    a. Relatório com o valor total do capital armazenado no banco.

Restrições:

Toda operação bancária (Saque, depósito e transferência) deverá ser registrada em um arquivo de texto de saída que armazena as operações realizadas durante aquela execução do sistema.

Todo relatório gerado deve ser registrado em um arquivo texto de saída individual.

O sistema deve realizar ao menos um tratamento de erros personalizado.

Sugestão: Caso um cliente tente realizar um depósito com valor indevido (valores negativos)

Requisitos - UML:

Assim como foi dito em aula, serão incluídos alguns diagramas de UML na avaliação final. Os diagramas obrigatórios constituem 8 pontos da nota total do projeto, que é 60 pontos. Já os diagramas opcionais valerão **10 pontos extras além dos 60 pontos dos elementos obrigatórios**, ou seja, se você fizer eles, poderá compensar a perda de nota no projeto.

Ex.: Se você tirar 45 no projeto em Java, tirar 8 nos diagramas obrigatórios e 10 nos diagramas opcionais, você ainda sim ficará com 60 pontos na sua nota final do projeto.

Diagramas Obrigatórios (8 pontos)

- Diagrama de classes do sistema desenvolvido pelo grupo;

Diagramas e Documentos Opcionais (10 pontos extras)

- Requisitos funcionais, não funcionais e regras de negócio (2 pontos);
- Diagrama de casos de uso (Somente o diagrama) (3 pontos);
- Descrição de 1 caso de uso com casos alternativos (Exemplo: login, saque, transferência) (2 pontos);
- Diagrama de atividade de 1 caso de uso (Exemplo: login, saque, transferência) (3 pontos);

Sugestões:

1. Antes de começar a programar, façam o **diagrama de classes**, porque servirá como um guia para o desenvolvimento de vocês e ajudará a prever possíveis dúvidas.
2. Também sugiro que façam o levantamento dos requisitos funcionais, não funcionais e regras de negócio antes de começarem a programar, pois servirá como um checklist do que já foi desenvolvido e o que ainda falta desenvolver.