

Alunos:

Arthur Kassick Ferreira

João Davi Martins Nunes

Lucas Henrik Miranda Souza

Yago Carvalho

Professora:

Ingrid Nunes

Técnicas de Construção de Programas 2019/1

Relatório do Trabalho Prático - Etapa 1

INTRODUÇÃO

Trabalho Prático (Etapa 1) consiste na evolução do projeto do Sistema Bancário, tanto em sua implementação em Java, como em seus diagramas (de classes e de sequência), outrora apresentado em aula pela professora. Nesta etapa, o grupo atua como *MAINTAINER* do projeto, adicionando algumas funcionalidades que incrementam o projeto original.

PONTOS DE MUDANÇA

Primeiro, mostramos como funciona o projeto originalmente.

1. O usuário, que possui uma conta no banco, deseja fazer um depósito.
2. Caso a solicitação seja feita de um caixa eletrônico, a agência e conta do cliente (usuário) são recuperadas da sessão, caso contrário (a solicitação seja feita de uma agência), o sistema solicita ao funcionário (usuário) agência e conta do cliente.
3. Sistema solicita a número do envelope, e o usuário informa.

4. Sistema solicita a quantidade a ser depositada, e o usuário informa.
5. Sistema credita o valor a ser depositado na conta.
6. Sistema registra a transação na conta.
7. Sistema exibe o status da operação.
8. Sistema retorna ao menu principal.

Agora, mostramos as mudanças, que consiste basicamente em um registro do estado do depósito, colocando-o em FINALIZADO ou PENDENTE, de acordo com os casos a seguir (mudanças nas etapas 5, 6 e 7):

5. Caso a solicitação seja feita em uma agência,
 - a. Sistema credita o valor a ser depositado na conta.
 - b. Sistema registra a transação com estado FINALIZADA.
6. Caso a solicitação seja feita em um caixa eletrônico e a quantidade a ser depositada seja inferior ou igual à R\$100,00,
 - a. Sistema credita o valor a ser depositado na conta.
 - b. Sistema registra a transação com estado PENDENTE.
7. Caso a solicitação seja feita em um caixa eletrônico e a quantidade a ser transferida seja superior a R\$100,00,
 - a. Sistema registra a transação com estado PENDENTE na conta de origem.

O usuário também deverá obter mais um recurso: poder CONFIRMAR DEPÓSITO, na qual uma funcionalidade que habilitaria-o de visualizar os depósitos pendentes. Nesta nova funcionalidade, o usuário poderá confirmar ou recusar os depósitos via sistema, alterando o saldo da sua conta.

DIAGRAMAS DE CLASSE

- Diagrama de Classe Business/Domain

A interface *AccountOperationService* (e a sua implementação) teve a adição de outro método *public Deposit deposit*, utilizando-se a técnica de reusa de sobrecarga com diferentes assinaturas, na qual *String status* e *boolean creditEnable* foram adicionados. Tais parâmetros servem para controlar o status do depósito e o seu crédito disponível. Mais 2 métodos foram incrementados nas classes: *getDepositByStatus* e *updateDeposits*. O primeiro método retorna uma lista ordenada de acordo com os depósitos existentes; o segundo, atualiza o status de um certo depósito existente na conta.

Na classe *CurrentAccount*, o mesmo método *public Deposit deposit* foi feito (explicação acima). O método *updateDeposit* também foi implementado porém apenas com 3 parâmetros, pois já possui acesso à conta. Também alteramos a visibilidade do método *depositAmount* de privado para público.

- Diagrama de classe Domain

A classe *Deposit* foi adicionado um atributo estático *BigDecimal verification_amount*, que é a quantia a ser verificada. Como consequência das modificações nas classes anteriores, o construtor foi alterado, adicionando *String status* como parâmetro.

Na classe abstrata *Transaction* é onde é implementada as modificações decorrentes do incremento do status do depósito. Um novo método construtor foi feito, adicionando o parâmetro do status, pois, se mexessemos no original, podíamos causar um erro no programa, em vista que poderiam haver outras classes e métodos que usassem o construtor original. Colocamos o *getter* e o *setter* do parâmetro *String status*. Um atributo estático *status* foi adicionado, que coloca em pendência os status dos depósitos do programa, além de mais 3 atributos públicos que irão certificar o status consequentes: *pending_status*, *accepted_status* e *refused_status*.

OBSERVAÇÕES

Diagramas no final do pdf seguem a sequência mostrada no relatório: Diagrama Business/Domain, Diagrama Domain e Diagrama Command.