

Programação Orientada a Aspectos

Cristiano de Faveri

1. Objetivos

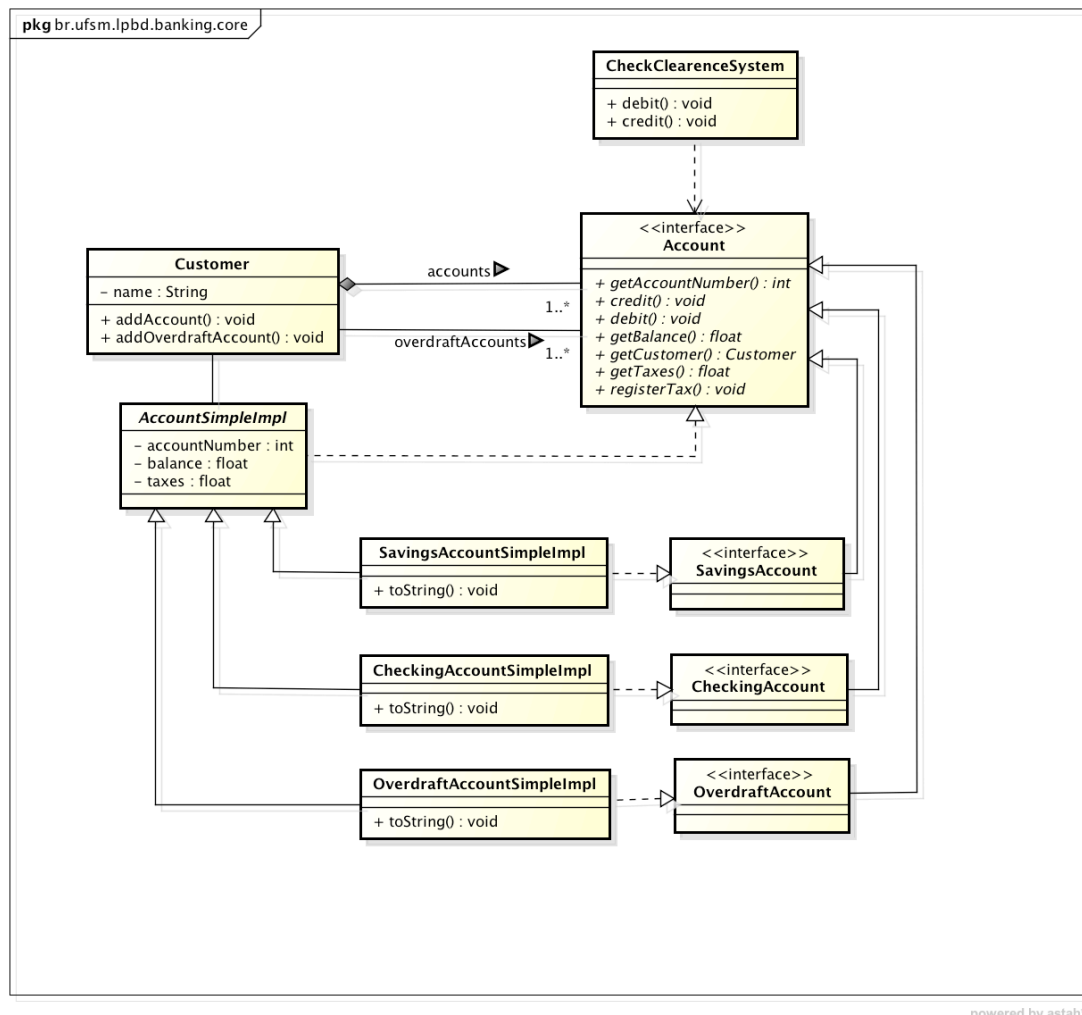
O objetivo dos exercícios é fixar conceitos de AOP através da implementação de interesses utilizando o AspectJ. Utilizaremos como base, um simples sistema de transações bancárias (AOPBanking), onde Aspectos serão implementados para atenderem a determinadas regras de negócio.

2. AOP Banking

O AOP Banking é um sistema que permite transações simples de crédito e débito em contas correntes e contas poupança. O cliente do banco pode possuir N contas correntes e N contas poupança, cada uma identificada com um número único. O cliente também possui pelo menos uma conta de empréstimo que possui um saldo determinado pelo banco para ser utilizado sempre que o saldo de conta corrente ou poupança não for suficiente para cobrir um cheque. Pela política do banco, débitos realizados pelo ATM não podem usufruir de empréstimos, somente quando compensados por cheques. Portanto, se um cliente realiza um saque em um ATM e não possui fundos na conta, o sistema não permite que a operação seja realizada. Porém se o débito for realizado pela compensação de um cheque, então o mecanismo de empréstimo entra em ação. Quando um empréstimo é realizado, uma taxa de 1% sobre o valor do empréstimo é cobrada e armazenada na conta para ser processada por outro sistema ao final de um período (mês por exemplo). Contas corrente e de poupança devem sempre manter um saldo de segurança de R\$100,00 para cobrir eventuais taxas bancárias.

3. Projeto OO

O cliente do banco é representado pela classe Customer e tem associado contas correntes (CheckingAccount) e/ou contas poupança (SavingAccount). A interface OverdraftAccount representa contas de empréstimo. As classes *impl são implementações das respectivas interfaces e descendem da classe abstrata AccountSimpleImpl que possui grande parte dos serviços de movimentação de contas.



4. Instalação e configuração do projeto

O projeto OO pode ser baixado e instalado como um projeto Eclipse. Caso prefira outra IDE, crie o projeto manualmente e importe os fontes e bibliotecas para o mesmo. O projeto traz um conjunto básico de testes executado por JUnit. Acrescente os test-cases que julgar necessário.

4.1 Bibliotecas necessárias

As bibliotecas necessárias estão na pasta `libs` do projeto, com exceção de AspectJ que deve ser baixado do site www.eclipse.org/aspectj.

Serão necessárias as libs junit-4.10-jar , aspectjrt.jar para desenvolvimento e execução do projeto.

- a) Faça download do projeto :
<https://sites.google.com/a/inf.ufsm.br/npp/AOPBanking.zip?attredirects=0&d=1>
- b) Copie-o para seu workspace do eclipse
- c) Faça as configurações necessárias para o projeto ser executado, caso seja necessário (build-path, aspectj, etc.)
- d) Implemente os aspectos no pacote br.ufsm.lpbdbanking.aspect

5. Lista de exercícios

- 5.1 Implemente um Aspecto que notifique o administrador do sistema sempre que um saque maior que R\$ 10000,00 for realizado em qualquer tipo de conta. A notificação deve conter o número da conta e o valor sacado. (Simule a notificação através de um System.out.print...)

Sugestão de implementação :

- a) Criar pointcut que capture chamadas de Account.debit,
 - b) Colete informações de Account e amount.
 - c) Implemente um advice before com a regra de notificação.
- 5.2 Implemente um Aspecto de logging para cada transação realizada em Account. Para cada operação realizada, antes da mesma ser realizada, deve-se imprimir (sysout) o nome do método, o número da conta, a quantia a ser transacionada. Ao final da transação deve-se imprimir o valor do saldo da conta.

Sugestão de implementação :

- a) Criar pointcut que capture execuções de Account.debit e Account.credit.
 - b) Colete informações de Account e amount.
 - c) Implemente advice before e advice.after com informações da transação. Estude a variável thisJointPointStaticPart para obter informações da operação executada.
- 5.3 Implemente um Aspecto que garanta que as transações de débito não permitam que o saldo fique menor que o limite de segurança de R\$ 100,00. Caso isso ocorra, uma exceção do tipo InsufficientBalanceException deve ser sinalizada.

Sugestão de implementação :

- a) Criar pointcut que capture chamadas de Account.debit e que não sejam de OverdraftAccount.
- b) Colete informações de Account e amount.
- c) Implemente advice around que verifique a violação da regra e execute ou não a operação de débito (proceed(..)).

5.4 Implemente um Aspecto que garanta a seguinte regra : Se um cheque for compensado (conta corrente ou poupança) e não houver saldo suficiente (abaixo do limite de segurança), porém houver saldo suficiente na conta de empréstimo (overdraft), o valor do saque é transferido para a conta corrente ou poupança para que a operação de saque seja realizada. Se não houver saldo suficiente na conta de empréstimo a exceção `InsufficientBalanceException` deve ser sinalizada. Atente-se que o saldo suficiente para a transação deve considerar o limite de segurança de R\$100,00. Ao transferir valores da conta de empréstimo para a conta corrente ou poupança o limite de segurança deve ser assegurado. Por exemplo, se houver R\$500,00 na conta de empréstimo e R\$200,00 na conta corrente e um cheque de R\$300,00 for compensado, será feito um saque de R\$200,00 da conta de empréstimo para cobrir o cheque com R\$100,00 e deixar R\$100,00 de limite de segurança na conta.

Sugestão de implementação :

- a) Criar pointcut que capture chamadas de Account.debit mas que não sejam de OverdraftAccount.
- b) Colete informações de Account e amount.
- c) Implemente advice around que verifique se há saldo suficiente (considerando o limite de segurança) para a transação. Em caso positivo, a transação é realizada. Caso contrário, é verificado se há saldo na conta de empréstimo. Se houver, é feita a transferência e a operação de débito ocorre normalmente. Se não houver saldo na conta de empréstimo, a exceção `InsufficientBalanceException` é sinalizada.

5.5 Implemente um Aspecto que adicione um valor de 1% do valor da transação referente a taxa bancária sempre que um empréstimo for realizado (transferência da conta overdraft para a conta corrente). As taxas serão processadas no final do mês por um outro sistema para então ter o saldo das contas corrente e poupança atualizado.

Sugestão de implementação :

- a) Criar pointcut que capture execuções da operação de transferência da conta de empréstimo para contas corrente ou poupança (Aspecto do item 5.4).
- b) Colete informações de Account e amount.
- c) Registre o imposto cobrado na conta (registerTax(..)).