

UNIVERSIDADE FEDERAL DO PARANÁ

CURSO DE ENGENHARIA ELÉTRICA

TE 353 - PROGRAMAÇÃO ORIENTADA A OBJETOS

Form de entrega do Trabalho 2 – Banco Object

Desenvolvedor

Estudante 1: Rian Marcos Sepulveda
GRR 1: 20196632
Estudante 2: Igor Leandro Bueno da Cruz
GRR 2: 20191826

Link para o vídeo

O ideal é que o vídeo não passe de 15 minutos, sendo necessário demonstrar o funcionamento das diversas funcionalidades constantes da especificação. Além de demonstrar o programa rodando, passar pelo código e comentar em linhas gerais a implementação das classes.

Caso seja em dupla, ambos precisam apresentar.

Ao subir o vídeo no YouTube, colocar novamente o nome na descrição e criar um índice de acesso ao vídeo para as seções fundamentais da apresentação na qual são demonstrado o funcionamento dos itens presentes nas rubricas de autocorreção (como criar o índice: https://youtu.be/mJAfTinzT_Y)

Importante:

- O vídeo precisa estar público
- Pode estar não listado: não aparece no canal, apenas quem tem o link acessa.
- Faça um teste com o link em um navegador não logado no YouTube (certifique-se que o vídeo está acessível!).

Link para o Vídeo: <https://youtu.be/DgbPVW9KT-4>

Autoavaliação

Para cada um dos itens da tabela das próximas páginas, escolha marcando um X àquele que mais se adapta ao resultado de sua implementação. Caso necessário, justifique a atribuição de algum item na tabela de comentários a seguir.

A soma total dos itens avaliados resulta em 11,5 pontos, anote abaixo a soma alcançada conforme sua justa e coerente autoavaliação.

A soma dos pontos é: 10,8

Comentários a respeito da implementação (o que gostaria de destacar de interessante, algo extra que foi feito e merece atenção, alguma dificuldade que foi resolvida/aprendida etc.)

Grade de autoavaliação do trabalho em relação a especificação

Funcionalidade criar conta corrente

- (X) 1.0 – Opera corretamente, lendo os atributos necessários do correntista, informando o número da conta, usando o construtor adequado e sua relação com a superclasse Conta e incluindo na lista de contas do Banco
- () 0.7 – Le os dados corretamente, porém não informa a conta ou não considera o mecanismo de herança
- () 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo).

() 0.0 – Não há esta possibilidade
<p>Funcionalidade criar conta poupança</p> <p>(X) 1.0 – Opera corretamente, lendo os atributos necessários do correntista, informando o número da conta, usando o construtor adequado e sua relação com a superclasse Conta e incluindo na lista de contas do Banco</p> <p>() 0.7 – Le os dados corretamente, porém não informa a conta ou não considera o mecanismo de herança</p> <p>() 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo).</p> <p>() 0.0 – Não há esta possibilidade</p>
<p>Funcionalidade efetuar depósito</p> <p>(X) 1.0 – Opera corretamente chamando algum método depositar da classe abstrata. Ou seja, a partir do número da conta informado apenas solicita o valor, pois independe se é Corrente ou Poupança, pegando a data corrente do sistema. A operação é corretamente armazenada na lista de operações da respectiva conta sendo de alguma forma identificada como um depósito</p> <p>() 0.7– Funciona, porém de outra forma levemente diferente, por exemplo: depositar está na subclasse, pergunta o tipo de conta, etc</p> <p>() 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo proposta)</p> <p>() 0.0 – Não há esta possibilidade</p>
<p>Funcionalidade efetuar saque</p> <p>(X) 1.0 – Opera corretamente chamando algum método sacar da classe abstrata. Ou seja, a partir do número da conta informado apenas solicita o valor, pois independe se é Corrente ou Poupança, pegando a data corrente do sistema. A operação é corretamente armazenada na lista de operações da respectiva conta sendo de alguma forma identificada como um saque</p> <p>() 0.7– Funciona, porém de outra forma levemente diferente, por exemplo: sacar está na subclasse, pergunta o tipo de conta, etc</p> <p>() 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo proposta)</p> <p>() 0.0 – Não há esta possibilidade</p>
<p>Funcionalidade aplicar correção</p> <p>(X) 1.0 – Opera corretamente perguntando a taxa de correção e aplica uma operação correção monetária em todas as contas do tipo Poupança sendo que a operação fica repertoriada na lista de operações da conta, com data e valor que foi calculado conforme o saldo vigente, sendo de alguma forma identificada como correção</p> <p>() 0.7– Funciona, porém de outra forma levemente diferente, por exemplo: não usa instanceof para verificar se é conta Poupança, não armazena algum atributo da operação (tipo, data ou valor), etc</p> <p>() 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo proposta) ou calcula sobre um saldo incorreto, que não é o vigente no momento da aplicação da correção</p> <p>() 0.0 – Não há esta possibilidade</p>
<p>Funcionalidade cadastrar PIX</p> <p>(X) 1.0 – Opera corretamente conforme especificado perguntando o CPF de um correntista, verificando se ele existe na lista de correntistas do Banco e se tem uma conta corrente, e em caso afirmativo adiciona este CPF à lista de correntistas do Banco que podem receber transações via PIX</p> <p>() 0.7 – Funciona, porém de outra forma levemente diferente, por exemplo, não possui uma lista específica de CPFs que aceitam PIX, ou usa alguma outra forma de acesso ou não verifica se é conta corrente etc</p> <p>() 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo)</p> <p>() 0.0 – Não há esta possibilidade</p>
<p>Funcionalidade efetuar PIX</p> <p>(X) 1.0 – Opera corretamente conforme especificado, ou seja, o sistema pergunta o CPF do destino da transferência e o valor, verifica se o destinatário está na lista de correntistas autorizados, e em caso afirmativo efetua retirada na origem usando algum método efetuar PIX (da classe ContaCorrente) e o depósito no destino chamando algum método receber PIX (da classe ContaCorrente). As respectivas operações de saída da origem e de entrada no destino ficam repertoriadas na lista de operações das contas envolvidas</p> <p>() 0.7 – Funciona, porém de outra forma levemente diferente, por exemplo, não possui uma lista específica de CPFs que aceitam PIX, e ou tem algum pequeno problema do registro de ambas operações nos históricos das contas envolvidas, não verifica se é conta corrente etc</p> <p>() 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo)</p> <p>() 0.0 – Não há esta possibilidade</p>
<p>Funcionalidade consultar extrato</p> <p>(X) 1.0 – Opera corretamente conforme especificado, ou seja, a partir do número da conta fornecido pelo usuário (independe de ser Corrente ou Poupança) exibe o extrato completo de todas as operações efetuadas na conta, mostrando o saldo após cada operação (ou ao final de cada dia que teve operação(ões)). Mostrar em cada linha do extrato a data, o tipo e o valor, consolidando o saldo (próximo ao que você obtém quando pega um extrato de sua conta no caixa automático de seu banco)</p> <p>() 0.7 – Funciona, porém de outra forma levemente diferente, por exemplo, não identifica algum tipo de operação, alguma das operações (depósito, saque, pix in, pix out, correção) não estão operando corretamente, a data ou saldo estão com algum pequeno problema, etc</p> <p>() 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo)</p> <p>() 0.0 – Não há esta possibilidade</p>

Emprego de HERANÇA e classe ABSTRATA

(X) 1.0 – Há uma classe Conta abstrata conforme especificado e duas subclasses concretas compatíveis com Conta Corrente e Conta Poupança, os construtores estão corretamente relacionados, dados encapsulados, lista de operações e funções de depósito e saque estão

na superclasse () 0.5 – O modelo de classes empregado foge um pouco da especificação, os métodos saque e depósito não estão na superclasse OU há algum problema com o uso de herança () 0.0 – Não há emprego de Herança ou de superclasse abstrata
Emprego de INTERFACES (X) 1.0 – Há uma interface Pix que é implementada pela classe responsável pela conta corrente E uma interface Remunerada que é implementada pela classe responsável pela conta poupança. Os métodos listados são corretamente implementados nas classes concretas. () 0.5 – As interfaces projetadas diferem levemente do proposto ou algum método em particular de uma delas não está totalmente implementado (efetuar PIX ou receber PIX ou aplicar correção) () 0.0 – Não há emprego de Interfaces
Atividades opcionais bônus (+ 1,5 pontos)
O código está documentado com tags JavaDoc e foi gerada a documentação html do sistema (X) 0.5 – Sim, todas as classes e métodos estão corretamente documentados com tags JavaDoc () 0.2 – As classes estão parcialmente documentadas (faltam algumas classes ou métodos), ou a documentação foi feita apenas com comentários tradicionais, sem uso de tags JavaDoc () 0.0 – Não há documentação com comentários ou tags
O sistema está persistindo os dados em disco usando serialização ou outro formato de gravação () 0.5 – Sim, o sistema salva suas estruturas de dados em disco quando de uma operação e abre o arquivo corretamente quando é iniciado, recuperando os dados persistidos. A funcionalidade foi demonstrada no vídeo de apresentação do trabalho (x) 0.0 – Não há persistência
O sistema emprega alguma classe de exceção criada para o contexto () 0.5 – Sim, o sistema tem uma ou mais classes de exceção programadas (por exemplo, subclasses de RuntimeException), para recusar algum parâmetro incorreto, não aceitar algum saque sobre saldo incompatível etc. A(s) classe(s) está(ão) apresentada(s) no vídeo do. Trabalho (não se trata aqui do uso de try-catch para manipulação de arquivos, e sim da criação de classe específica) (x) 0.0 – Não há emprego de classes de exceção criadas para o sistema
Atividades opcionais extras
A implementação possui alguma <u>funcionalidade extra</u>, se SIM, comente qual(is) neste espaço e no vídeo que serão avaliadas como bonificação: Sim, validação do CPF