Fatec São Caetano do Sul – Antonio Russo

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - AMS		
AVALIAÇÃO OFICIAL	**DISCIPLINA: TÉCNICAS AVANÇADAS DE PROGRAMAÇÃO**	**NOTA**
DATA: 05/06/2024 ☐ N1 ☒ N2 ☐ N3 ☐ N4	**TURMA: ADS - AMS** **PROFESSOR: FERNANDO TONIOLLI**	
ALUNO:	**RA:**	
Data devolução para o aluno: _____/_____/_____	*Ciência do Aluno (vista de prova)*	
INSTRUÇÕES: **INSTRUMENTO DE AVALIAÇÃO: Projeto Sistema Bancário – Parte 2** **CONDIÇÕES: Trabalho individual** **TEMPO MÁXIMO DE DURAÇÃO: Postar no Teams até 19/junho/2024**		

1) Projeto Sistema Bancário - Parte 2 – Conta Corrente e Poupança - Polimorfismo – Override - Abstract

Prazo de entrega: até 19/junho/2024

A Parte 2 do Projeto Sistema Bancário é uma continuação do código feito por vocês na Parte 1 do Projeto. Dessa forma, certifique-se que seu código atende aos requisitos representados pela UML abaixo. Caso não atenda 100%, faça os devidos ajustes em seu código da Parte 1 antes de fazer código referente às especificações da Parte 2.

Pudemos observar na Parte 1 do Projeto um exemplo de Polimorfismo com o método sacar quando não se sabe ainda se é uma conta corrente, ou conta poupança. Este código funciona por causa do polimorfismo.

Nosso chefe nos informou que houve uma mudança na regra de negócio referente ao saque: no caso do saque em Conta Corrente haverá a cobrança de uma tarifa de R$0,20 por saque.

Altere o código JAVA criado na Parte 1 do Projeto para implementar essa mudança na regra de negócio.

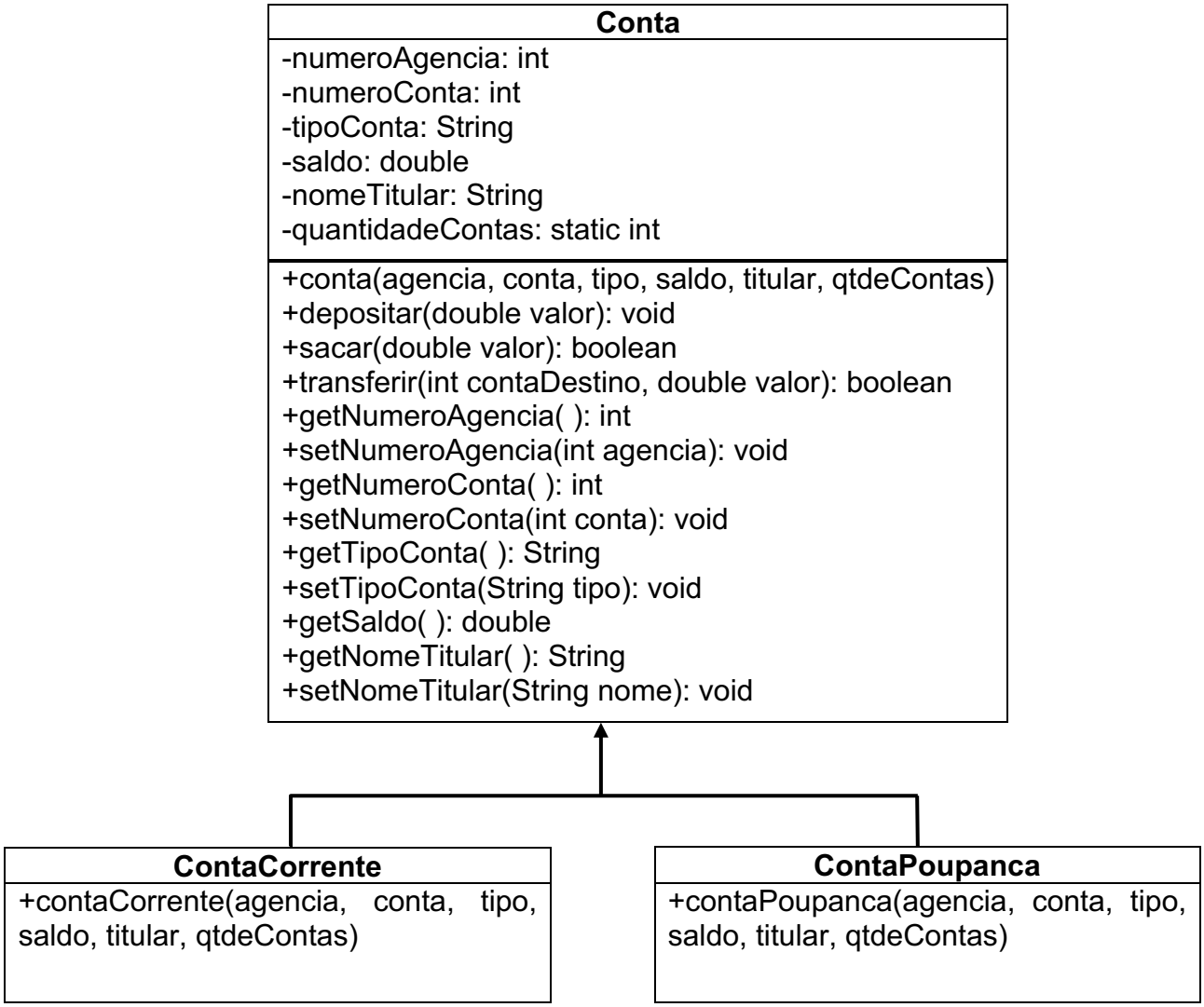
Caso não tenha definido a Classe Conta, na Parte 1 do Projeto, do tipo “abstract”, faça isso agora, nessa Parte 2 do Projeto quando estiver ajustando o código JAVA devido a mudança da regra de negócio.

Observe que ao transformar a Classe Conta para “abstract”, todo o código continua funcionando, os atributos, os métodos, os construtores, tudo continua compilando sem erros, a única exceção é que para de funcionar o instanciamento de objetos a partir da Classe Conta pois Conta virou uma classe abstrata.

Para fixar melhor o conceito de “abstract”, vamos alterar o método depositar para abstrato (mesmo que não haja uma regra de negócio que demande isso). Percebam que imediatamente o eclipse acusa erro na compilação desse método (“Abstract methods do not specify a body”). Alterem o código de forma a tornar o método depositar como abstrato e não dar erro de compilação e a classe de teste continuar a funcionar tal como funcionou na Parte 1 do Projeto.

Atenção ao atributo saldo, definido na classe Conta com private. O que vocês podem fazer para torná-lo visível nas classes filhas (Conta Corrente e Conta Poupança) caso não se queira usar o método getSaldo? Não vale alterar esse atributo para “public”.

Antes de iniciar os ajustes no código JAVA da Parte 1 do Projeto, certifique-se que o ponto de partida, seu código, atende aos requisitos representados pela UML a seguir.



Altere a classe Teste para mostrar que tudo continuou funcionando após as alterações.

Atenção: Ao postar suas classes no Teams, NÃO compacte os arquivos gerando arquivo.RAR ou .ZIP para não inviabilizar sua abertura no Teams.