



Universidade de Brasília

Departamento de Ciência da Computação

Linguagens de Programação – 2020/2 – Turma A, Prof. Vander Alves

Trabalho 3

Entrega: até 20/05 às 23:59h. Após isso, desconto de 1,0 ponto por dia de atraso até o dia 23/05.

Favor seguir as seguintes instruções:

- Forma de entrega: tarefa no Aprender 3 através do envio do arquivo **Trabalho3.rar** contendo apenas os arquivos-fontes (.java).
- O trabalho pode ser feito em grupo de no máximo 3 alunos. A identificação do grupo deve ser feita como comentário em Java no início do arquivo **InterfaceTextual.java**
- Ler os critérios de avaliação após o enunciado da Questão 3.

Bom trabalho!

Questão 1 (3,00 pontos)

Evolua a aplicação base (arquivos fontes em **aplicacaoBase.rar**) de forma a contemplar dois novos tipos de conta: **ContaBonificada** e **Poupanca**. Defina, portanto, nesta questão, duas novas classes: **ContaBonificada** e **Poupanca**, ambas subclasses de **Conta**, conforme discutido em aula. De acordo com o princípio de *Information Hiding*, atributos devem ser declarados com o modificador de acesso **private**, e métodos e construtores com o modificador **public**.

Questão 2 (3,00 pontos)

A aplicação deve permitir a criação de **ContaBonificada** e **Poupanca**, além da própria **Conta**. A interface perguntará ao usuário qual tipo de conta ele deseja criar. *Dica:* estude trecho de código comentado no arquivo **InterfaceTextual.java**.

Questão 3 (4,00 pontos)

A aplicação deve permitir chamar as seguintes operações:

- *renderBonus* para uma conta bonificada;
- *renderJuros* para uma poupanca.

Para isto, deverá ser fornecido o número da conta. *Dica:* estude trecho de código comentado no arquivo **InterfaceTextual.java**. Além desse arquivo, você precisará alterar os arquivos **Fachada.java** e **CadastroContas.java**. Note que os métodos **renderBonus** e **renderJuros** já devem ter sido implementados nas classes **ContaBonificada** e **Poupanca**, respectivamente, na sua resposta à Questão 1.

Observação: a aplicação deve estar estruturada de acordo com a arquitetura em camadas, conforme apresentado em aula, e a interface textual da aplicação deve ser conforme a seguir:

*** Aplicação bancária ***

Operações disponíveis:

- 1- Criar cliente
- 2- Consultar cliente
- 3- Atualizar cliente
- 4- Remover cliente
- 5- Criar conta (Conta, Poupança, ou Conta Bonificada)
- 6- Consultar conta
- 7- Atualizar conta
- 8- Remover conta
- 9- Creditar em conta
- 10- Debitar de conta
- 11- Transferir entre contas
- 12- Render bonus sobre conta bonificada
- 13- Render juros sobre uma poupança
- 14- Exibir os dados da conta de um determinado cliente
- 15- Exibir os dados de todas as contas
- 16- Exibir os dados de todos os clientes
- 17- Sair da aplicação

Favor escolher uma opção: _

Critérios de avaliação

- Casos de testes da equipe de avaliação (monitores e professor): 60%
- Conformidade à arquitetura em camadas: 20%
- Inspeção do código fonte: correção, robustez, legibilidade: 20%
- **Atenção:** terão nota zero trabalhos plagiados ou que não sejam compiláveis por problemas de entrada (p.ex. erros léxicos e de sintaxe) e formatação.