

Não é permitida a desistência após o aluno ter acesso à prova e é necessário esperar pelo menos 30 minutos para entregar a prova. Só serão consideradas as respostas que estiverem na folha pautada e algoritmos sem indentação serão desconsiderados.

Utilize os dados abaixo para a elaboração da sua prova. Suponha um sistema bancário com as seguintes classes básicas:

- i) Conta, com os atributos número (int), saldo (double) e senha (String) e os seguintes métodos getNumero, setNumero, getSaldo, setSenha e boolean validaSenha(String).
- ii) Poupanca (que herda de Conta) com o atributo diaBase (int) e o método rende(double juros, double correcaoMonetaria).
- iii) Cliente com os atributos nome (String), cpf (String) e CadastroContas (que utiliza um RepositorioContasArray)

Questão 1) (1,0 ponto) Sabendo que existem dois tipos de repositórios para cada uma das classes básicas (array e banco de dados) e que a interface do sistema será do tipo console, mostre a visão arquitetural de um sistema bancário utilizando 3 camadas (faça o desenho contendo todas as classes e interfaces necessárias).

Questão 2) (1,5 ponto) Implemente a classe Conta (só precisa declarar os atributos e os métodos validaSenha que recebe uma senha como parâmetro e retorna true ou false dependendo se a senha foi ou não correta e setSenha que altera a senha da conta. OBS: se o método validaSenha for chamado três vezes consecutivamente com uma senha errada esta senha ficará invalidada e sempre o método validaSenha deve retornar false (mesmo passando depois a senha correta), neste caso, é necessário alterar a senha com o setSenha para o funcionamento do validaSenha voltar ao normal.

Questão 3) (1,0 ponto) Implemente as classes ExcecaoContaJaCadastrada (que é utilizada quando tenta-se cadastrar outra conta com o mesmo número) e ExcecaoContaInvalida (quando se tenta cadastrar uma conta nula, com número zero ou negativo ou com saldo negativo).

Questão 4) (1,5 ponto) Implemente a classe repositorio de contas array (só precisa declarar os atributos, criar o construtor padrão que inicializa um vetor de tamanho 50 e implementar o método cadastrar que recebe uma conta como parâmetro. Se o vetor estiver cheio, crie um novo e passe todos as contas cadastradas para este novo vetor e jogue fora o vetor antigo.

Questão 5) (1,0 ponto) Implemente a classe controlador de contas (só precisa declarar os atributos, criar o construtor que recebe um repositório de contas como parâmetro e implementar o método cadastrar que recebe uma Conta como parâmetro).

Questão 6) (1,0 ponto) Implemente a classe Fachada deste sistema como um Singleton e com o construtor padrão que cria todos os cadastros utilizando repositórios do tipo banco de dados (só precisa declarar os atributos, criar o construtor, implementar o método necessário para a classe ser Singleton e fazer o método que cadastra uma conta recebida como parâmetro).

Questão para Substituir o Projeto) (3,0 pontos) Implemente a classe Cliente completa (sem os get/set), inclusive o método cadastrarConta. Adicione nas classes fachada, controlador de contas e no repositório de contas array o método remover que recebe o número de uma conta como parâmetro. Lembre-se de criar também qualquer classe extra necessária.

Boa Prova!