

CURSO SUPERIOR DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS – ADS

AValiação Oficial	DISCIPLINA: Programação Orientada a Objetos – POO	NOTA
DATA: 26/11/2020 <input type="checkbox"/> N1 <input checked="" type="checkbox"/> N2 <input type="checkbox"/> N3	TURMA: ADS – VA4 PROFESSOR: M. Sc. Viotti _____	
ALUNO: Felix Petiz Bonilho		RM: 1680481911002
ALUNO: Caroline Bognar da Silva		RM: 1680481911053
INSTRUMENTO DE AVALIAÇÃO: Prova Prática CONDIÇÕES: Individual TEMPO MÁXIMO DE DURAÇÃO: 1,40' TEMPO MÍNIMO DE DURAÇÃO: 50'		Visto do Aluno (após correção)

Orientações

- Qualquer comprovação, parcial ou total, de cópia de qualquer questão da prova, acarretará a anulação total da prova, recebendo como nota 0.0 (zero).
- Não poderá existir erros de compilação. Caso exista a questão será desconsiderada da prova
- A prova toda deverá ser confeccionada em um único projeto. Faça a divisão das questões através da criação de “Pacotes” dentro de projeto, e nomeie cada pacote com o número da questão. Exemplo: EX_01
- Caso necessário, duplique suas classes, com isso você poderá possuir classes independentes para cada questão.

Tratamento de Exceção

1. (2,0 pontos) Crie duas classes de exceção, uma verificada (VerificadaException) e outra não verificada (NaoVerificadaException). Crie também a classe demonstrada abaixo. Os métodos devem seguir os seguintes critérios:

- a. `valorEmEstoque()`: retorna a quantidade em estoque multiplicado pelo preço. Este método pode gerar uma exceção verificada, caso a quantidade de estoque esteja zerada.
- b. `margemLucro()`: Deve devolver a diferença entre o preço do produto e o mesmo preço acrescido de 10%. Esse método pode gerar uma exceção não verificada, caso o preço esteja zerado

Crie um método principal que instancie um Produto e atribua valores para cada um dos atributos. Ao final você deverá exibir o Valor em Estoque e a Margem de Lucro. Sempre utilizando os métodos criados.

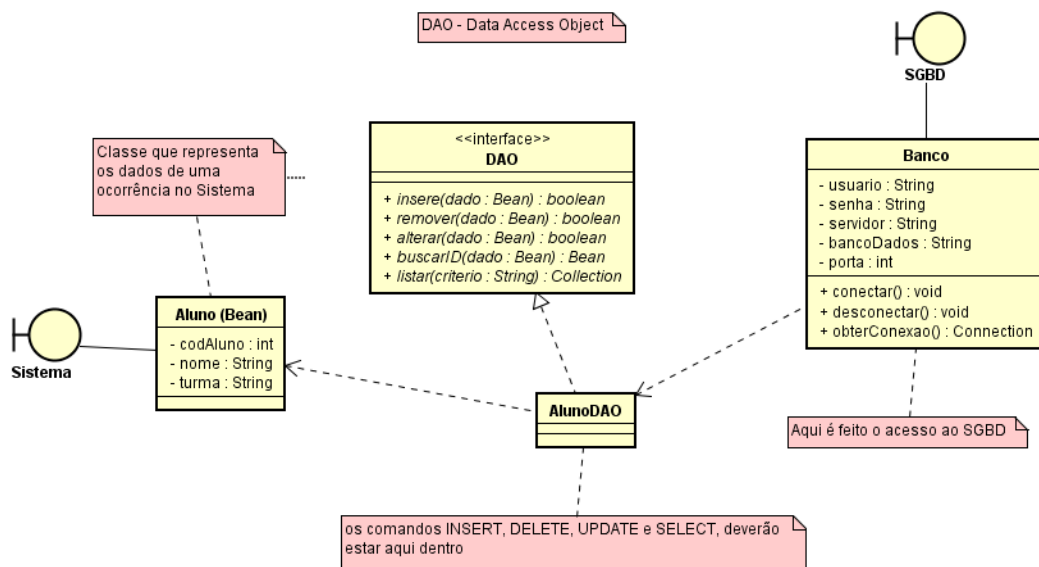
Produto
- descricao : String
- qtdEstoque : int
- preco : float
+ valorEmEstoque() : float
+ margemLucro() : float

Uso da ComboBox

2. (3,0 pontos) Crie um formulário conforme o descrito abaixo, utilize o modelo de classe apresentado na questão 1 e faça:
 - a. (0,5) Criação do formulário, com todos os nomes corretos de acordo com o padrão estabelecido em aula.
 - b. (0,5) Configuração da classe Produto para que seja exibido na combo somente a descrição do mesmo, e que a comparação entre objetos Produto ocorra comparando-se o campo 'Descrição'
 - c. (1,0) Ao clicar no botão "Inserir na Combo" um objeto Produto deverá ser inserido dentro da combo. Não será permitido que exista Produtos repetidos. Dica: Você pode utilizar o método "getIndexOf()" que faz uma procura. Lembre-se, uma Combo utiliza uma coleção dentro dele, e já foi ensinado como se faz uma pesquisa dentro de uma coleção.
 - d. (1,0) Ao clicar no botão "Apresentar Dados da Combo" apresente em um JOptionPane() todas as informações do objeto selecionado.

Acesso a Dados

3. (5,0 pontos) Crie um formulário conforme o descrito abaixo, use também o modelo DAO abaixo:



Dados de Alunos

Matrícula

Nome:

Turma

- (0,5) Crie um formulário de acordo com o exibido acima
- (0,5) Crie os métodos auxiliares para movimento de informações da tela para o objeto e vice versa, limpeza de campos entre outros
- (1,5) Crie a interface DAO e implemente a classe AlunoDAO
- (0,5) Crie a classe Banco para acessar seu banco de dados chamado Prova, com a tabela Aluno, de acordo com a classe exibida acima.
- (1,0) Ao clicar no botão "Incluir" você deverá fazer:
 - Validação de dados: Os campos devem estar preenchidos
 - Fazer a inclusão dos dados na tabela do banco de dados
- (1,0) Ao clicar no botão "Pesquisar", faça a pesquisa do aluno no banco, procurando por sua matricula, caso a encontre, exiba os dados nos campos, caso contrário exiba uma mensagem de erro.