

# Prova 3

Algoritmos e Programação Orientada a Objetos II — 2017

## Instruções para a realização da prova:

1. A prova contém **3 questões práticas**, totalizando **12 pontos**;
2. A prova é **individual** e **sem consulta**;
3. **Não altere a disposição do mobiliário da prova**;
4. **Coloque seu celular** em cima do gabinete, desligado. Caso o professor ouça o celular vibrando ou tocando durante a aula, o aluno receberá **nota zero**;
5. **Faça a prova em silêncio**; não converse durante a prova;
6. **Não tente plagiar a prova do(a) seu(sua) colega**, você pode prejudicar você e seu(sua) colega;
7. **Não é necessário verificar a entrada**; isto é, se seu programa solicita que o(a) usuário(a) informe um número inteiro e o(a) usuário(a) informa uma letra ou qualquer outra coisa diferente de um número, seu programa pode ter qualquer comportamento inesperado;
8. Dentro da pasta *Documentos*, **crie uma pasta com seu nome completo** e coloque todos os arquivos da sua prova dentro dessa pasta.
9. Utilize um editor de textos comum ou a IDE Eclipse.

BOA PROVA!

## Questões

1. (4.0) Uma determinada livraria precisa de um sistema para catalogar itens colecionáveis (livros, CDs, DVDs e revistas). O objetivo deste sistema é manter os itens colecionáveis, organizados por tipo. O sistema deve permitir cadastrar os dados comuns e os específicos de cada tipo de item. Os dados comuns são: identificação única, nome, data de aquisição e lista de autores.

Para os livros é importante manter o nome da editora e o ano de publicação. Já para os CDs, é interessante manter o gênero musical e a identificação das faixas de áudio (nomes das músicas). Para os DVDs é importante armazenar o tipo (musical, filme ou dados), e uma descrição geral. Por fim, para as revistas é interessante manter o ano de publicação, o volume, a editora e os principais assuntos tratados.

Modele o problema acima utilizando classes e implemente uma classe que teste que forneça as seguintes opções de menu:

- (a) Cadastrar um livro;
- (b) Cadastrar um CD;
- (c) Cadastrar um DVD;
- (d) Cadastrar uma revista;
- (e) Listar todos os itens cadastrados.

Neste exercício você deve utilizar pelo menos uma variável do tipo *Enum* e pelo menos uma classe abstrata. Além disso, o item 5 da classe de testes deve ser implementado utilizando polimorfismo (dica: utilize o método *toString* e uma coleção com objetos da classe base).

2. (4.0) A Receita Federal está atualizando seu cadastro de CPFs e, para isso, precisa de um sistema capaz de cadastrar novos CPFs e atualizar as informações dos contribuintes já cadastrados. Além do CPF, uma pessoa possui nome, data de nascimento e endereço.

Utilizando orientação a objetos, crie um sistema capaz de:

- Buscar as informações de um contribuinte com base no seu CPF;
- Cadastrar um novo contribuinte com todas as suas informações. Caso o CPF informado já esteja cadastrado, você deve emitir uma mensagem de erro;
- Atualizar os dados de um contribuinte com base no seu CPF;
- Listar todos os cpfs cujo cadastrado, em algum momento, foi duplicado. Liste os CPFs *SEM REPETI-LOS*.

Neste exercício, as únicas coleções genéricas permitidas são a *Map* e a *Set*.

3. (4.0) Implemente uma classe chamada *VetoriusInfinitus*< *T* >, capaz de modelar um vetor 'infinito' de tipo *T* utilizando um vetor. Sua classe deve fornecer métodos para adicionar um elemento no final (caso não caiba no vetor interno, você deve redimensioná-lo), acessar um elemento de uma determinada posição (caso essa posição não exista, lance a exceção não verificada *ArrayIndexOutOfBoundsException*) e retornar a quantidade de elementos inseridos no vetor. Implemente uma classe com método *main* que crie uma instância de *VetoriusInfinitus* e teste todos os seus métodos.