



### Trabalho de Implementação

#### Observações

- O trabalho deve ser realizado em grupos de, no máximo, 3 pessoas
- Entregar o modelo de classes do sistema em um arquivo DOC
  - Utilize alguma ferramenta UML para criar o modelo
- Entregar o trabalho via ambiente Moodle
  - Entregar um arquivo compactado contendo somente as **classes** do projeto e o **modelo**
  - Nome do arquivo deve seguir o formato TOO.Aluno1.Aluno2.Aluno3.rar (ou zip)
- O grupo deverá apresentar o trabalho ao professor no dia **09/12/11**
- O sistema deve ser implementado utilizando-se linguagem Java
- **Data de entrega: 07/12/11 (até as 23:00)**
  - Trabalhos entregues após a data não serão aceitos, em hipótese alguma

#### Locadora de Vídeo

Uma Locadora de Vídeo precisa ter controle sobre o cadastro de clientes e acervo de filmes disponíveis para locação. Desenvolva uma aplicação que permita criar um cadastro de clientes e de filmes, e acessá-los quando necessário.

- I) Crie uma classe Pessoa com os campos nome e dataNasc (da classe String),
  - a. Os campos devem ser inicializados pelo construtor.
- II) Crie uma classe Cliente, que estende Pessoa, e possua um campo String endereco e um campo do tipo int codigoCliente. O construtor da classe Cliente deve inicializar todos os seus campos. A classe Cliente também deve ter um campo Histórico da classe Vector. Este campo deve armazenar objetos da classe Aluguel. A classe Aluguel deve conter a data de locação, data de devolução e o código do Filme alugado. A classe Cliente deve ter um método adFilmeHist(), que recebe a data de locação, data de devolução e o código do Filme alugado, cria um objeto Aluguel com estas informações e adiciona o mesmo no Histórico.
- III) Crie uma classe Filme
  - a. Com campos para:
    - Código do Filme (caracteres alfanuméricos)
    - Título do Filme
    - Categoria (que pode ser Aventura, Ficção, etc.)
    - Quantidade (inteiro)
    - Alugados (inteiro)
  - b. Construtores (dois) para:
    - Inicializar todos os seus campos e
    - Receber como parâmetro um objeto String e inicializa o campo do Título do Filme.
  - c. Métodos
    - aloca, que não recebe parâmetros e acerta o campo Alugados. Caso todas as cópias estejam alugadas, deve levantar a exceção confirmada CopiaNaoDisponivelEx;
    - devolve, que não recebe parâmetros e acerta o campo Alugados. Caso nenhuma cópia tenha sido alugada, deve levantar a exceção NenhumaCopiaAlugadaEx.

#### IV) Implemente a classe Locadora

- a. Com campos para:
  - Cadastro de clientes (do tipo `java.util.HashMap`)
  - Cadastro de filmes (`java.util.HashMap`)
- b. Construtor para inicializar os campos
- c. Métodos
  - `cadastraCliente`: de retorno `void`, que recebe como parâmetro um objeto da classe `Cliente` e o armazena no campo `HashMap` correspondente. O código do cliente deve ser utilizado como chave.
  - `cadastraFilme`: de retorno `void`, recebe como parâmetro um objeto da classe `Filme` e o armazena no campo `HashMap` correspondente; O código do Filme deve ser usado como chave;
  - `alugaFilme`: recebe como parâmetros a referência a um objeto `Cliente` e a referência a um objeto da classe `Filme`. (as referências já devem ter sido validadas – obtidas através dos métodos `getFilme` e `getCliente` – abaixo). Chama o método `aloca` no objeto `Filme` e atualiza o histórico no objeto `Cliente` chamando `adFilmeHist`.
  - `String imprimeFilmes()`: Devolve uma `string` com a lista de filmes cadastrados, ordenados pelo título.
  - `String imprimeClientes()`: Devolve uma `string` com a lista de clientes cadastrados, ordenados pelo nome.
  - `Filme getFilme (int cód)`: Recebe o código do filme e obtém o objeto `Filme` da `HashMap` correspondente. Se o filme estiver cadastrado, deve gerar a exceção `FilmeNaoCadastradoEx`;
  - `Cliente getCliente (int cód)`: Recebe o código do cliente e obtém o objeto `Cliente` da `HashMap` correspondente. Se o cliente não existir na `HashMap`, deve gerar a exceção `ClienteNaoCadastradoEx`;

Obs.: Além dos métodos listados, verifique se as classes `Pessoa`, `Cliente`, `Filme` e `Locadora`, devem ter, para cada campo, um método `get<NomeDoCampo>()` que retorna o conteúdo do campo.

#### V) Desenvolver o programa principal, que deve ter os seguintes métodos:

- a. Cadastro de cliente
- b. Cadastro de filme
- c. Locação: que executa a locação de um filme para um cliente.

#### VI) Construção de Menu

- a. Deve ser construído um menu que possibilite ao usuário escolher entre (i) cadastrar clientes, (ii) cadastrar filmes, (iii) exibir o cadastro/lista de filmes e (iv) fazer uma locação. Será necessário ao grupo pesquisar a realização de **entrada de dados via teclado**. Não utilize nada relacionado a interface gráfica. Não será cobrado o armazenamento dos dados em arquivos em banco de dados.

OBS: Ao fazer uma locação, deve ser obtido o objeto `Filme` e `Cliente`. As exceções (filme não existe, cliente não cadastrado, cópia não disponível, etc.) devem ser tratadas. Quando uma locação for bem sucedida os dados devem ser escritos na tela e o histórico do cliente atualizado. Obs: no programa principal, o programador deve customizar a política da locadora. Ex: pode restringir o número máximo de filmes que um usuário pode alugar de uma só vez e o número de dias que um usuário pode ficar com a fita sem pagar novo aluguel.