Atividade APS – 2020.2
Professor Jefferson de Carvalho

Fazendo uso da linguagem JAVA, e em equipe de no máximo 4 alunos, implemente o seguinte programa:

1. Objetivo

Desenvolver um sistema de locação de filmes e música, com interface em console. O sistema será capaz de alugar os produtos aos clientes cadastrados, gerar multas de atraso, gerar boletos de pagamento, listar produtos, buscar produtos, cadastrar novos produtos e permitir três tipos de usuário: o cliente, o gerente e o operador do sistema.

2. Entidades

2.1 – Produtos

Os produtos disponíveis na locadora são Filmes, em Blu-ray, DVD, VHS e Músicas em CD e LP (discos de vinil).

Um Produto é uma Classe Abstrata que deve ter os seguintes atributos: String: codigo, String: título, String: gênero, boolean: locado. Além disso, um Produto declara um método abstrato chamado **double calcularDiaria()**.

Um Filme **é-um** Produto e tem os seguintes atributos: int: anoLancamento e int: duracao (em minutos). Filme também é uma classe abtrata.

DVDs, VHSs e Blu-rays **são** Filmes e são classes concretas. DVD tem um atributo boolean: arranhado e VHS um atributo boolean: cores; ou seja, um DVD pode estar arranhado ou não e um VHS pode ser a cores ou não. Um Blu-ray tem um atributo String[2]:idiomas que armazena quais os idiomas disponíveis para o Filme.

Musica **é-um** Produto. Assim como Filme, também é uma classe Abstrata. Tem como atributos String: autor e int: numFaixas.

LPs e CDs são filhos de Musica. LP tem como atributo boolean: raro e CD tem como atributos boolean: arranhado e boolean: duplo.

2.2 - Atores

Os atores envolvidos no sistema são: o Cliente, o Gerente, e o Operador do Sistema. Criaremos uma classe Pessoa para generalizar seus atributos.

A Classe **Pessoa** é abstrata e deverá ter os seguintes atributos: String: nome, int: matricula.

A Classe Funcionário é filha de Pessoa e deve ter os atributos String: login e String: senha.

A classe Cliente é filha de Pessoa mas Gerente e Operador de Sistema são filhas de Funcionário.

Um **Gerente** deve ter as seguintes ações:

- Adicionar um Cliente
- Adicionar um Produto qualquer
- Adicionar um Operador
- Listar os Clientes, Produtos e Operadores
- Procurar Cliente, Produtos e Operadores por matricula ou código

Um **Cliente** deve ter:

- Um atributo String: endereço
- Um atributo int: idade
- Um atributo char: sexo

Um **Operador de Sistema** deve:

- Fazer uma locação de um produto a um cliente
- Excluir uma locação
- Fazer baixa de uma locação, cobrando as diárias e multas, caso existam.
- Procurar produtos por codigo
- Procurar Clientes por matricula.

2.3 – Repositórios

Os repositórios servem para armazenar a nossa Base de Dados **em memória**. Vão existir 3 tipos de repositórios: o de Locações, o de Produtos e o de Pessoas. O aluno deve criar uma classe para cada Repositório e armazenar os objetos da seguinte forma:

1. No caso de Pessoas, usar uma **Hashmap** onde a chave é a matrícula da pessoa e o valor é o objeto em si.
2. No caso de Produtos, também usar um **Hashmap** onde a chave é o código do Produto e o valor é o objeto do tipo produto.
3. No caso de Locações, o Aluno deve criar uma classe intermediária chamada de **Locação**. Nesta Classe, serão armazenados os seguintes atributos:
    1. código do produto
    2. matrícula do cliente
    3. Data de saída (java.util.Date)
    4. Data de prevista de entrega (java.util.Date)
    5. Essa Classe deverá ter um método que calculará a multa caso ela exista.

    Feito isso, o aluno armazenará as Locações em um **ArrayList**.

Os repositórios terão funções em comum que são as de adicionar, remover e retornar um objeto em sua coleção. Fica a cargo do aluno o nome destas funções. Usem padrões de convenção de código.

Os repositórios estão relacionados com outras classes, por exemplo: Gerente tem acesso ao Repositório de Pessoas e Produtos mas não acessa o Repositório de Locações pois esse só é de interesse do Operador. O Operador, no entanto, acessa todos os repositórios.

3. A Interface

O sistema será implementado usando uma interface em Console, abusando da classe **Scanner**,

como os trabalhos anteriores. O aluno deverá, inicialmente perguntar qual perfil o usuário deseja fazer login. Veja o Exemplo:

****Bem-vindo, escolha um perfil abaixo:
1 – Gerente
2 – Operador de Sistema
3 – Sair
Digite a opção: _<ENTER>

Qualquer uma das opções levará a tela de Login

****Faça seu Login****
Digite seu login:_ <ENTER>
Digite sua senha:_ <ENTER>

O sistema verificará se o login e senha corresponde com um Gerente ou Operador. Caso dê tudo certo, as telas das opções será mostrada para o perfil desejado:

Para o Gerente:

****Olá Gerente <Nome do Gerente>*****
1 – Cadastrar Produto
2 – Cadastrar Cliente
3 – Cadastrar Operador
4 – Listar Produtos
5 – Listar Clientes
6 – Listar Operadores
7 – Procurar Produto
8 – Procurar Cliente
9 – Procurar Operador
10 – Sair
Digite a opção:_<ENTER>

Para o Operador

****Olá Operador <Nome do Operador>****
1 – Fazer locação
2 – Dar baixa em locação
3 – Excluir locação
4 – Procurar Produto
5 – Procurar Cliente
6 – Sair
Digite a opção:_<ENTER>

Gerente:

Nas telas de opção 1,2 e 3 pro Gerente, o sistema alimentará, via Scanner, todos os dados das classes correspondentes (Produto, Operador ou Clientes) usando o **sets.**
Nas telas de opção 4,5 e 6 pro Gerente, serão mostrados os dados referentes ao objeto escolhido um por um. Use um laço e imprima na tela as informações de cada objeto usando o método toString();
Nas telas de 7,8,9 pro Gerente, apenas peça a matrícula e mostre o Produto ou Pessoa correspondente caso ela **exista.** Para isto, o sistema deve acessar os Repositórios

Operador:

A opção “Fazer uma locação” do operador pedirá duas informações: a matrícula do cliente e o código do produto. Se o produto já estiver alugado, o sistema deverá informar que não é possível alugá-lo. Se o produto estiver livre, um objeto do tipo locação será armazenado no repositório de locações, contendo a matrícula do Cliente, o código do produto e a datas de saída e data prevista para entrega. A data de previsão de entrega será de **duas** diárias sobre a data de saída.

Ao dar baixa numa locação, o Operador irá também pedir as mesmas informações da operação anterior. O seu sistema deverá percorrer o Repositório de Locações para achar uma locação que contenha o código e matrícula do cliente (Dica, use o **equals** para comparar locações). Uma vez encontrada, o sistema gerará o valor do das diárias e as multas caso a data de entrega ultrapasse a data prevista.

O aluno fica livre em estabelecer qualquer valor para uma multa e diárias de locação.

4. Considerações Finais

1. As equipes serão formadas por no máximo **4 alunos**.
2. A atividade deve ser submetida no GITHUB de um dos integrantes da equipe, mas sintam-se livres de ter a atividade no GITHUB de todos os integrantes.
3. Ao enviar a atividade no SIPPA, enviem um documento TXT com o nome de TODOS os integrantes da equipe e o LINK do GITHUB da atividade (BASTA APENAS UM INTEGRANTE ENVIAR NO SIPPA).
4. Enviem no SIPPA para **A02.**
5. **PRAZO: 13/12, até 23:59**