

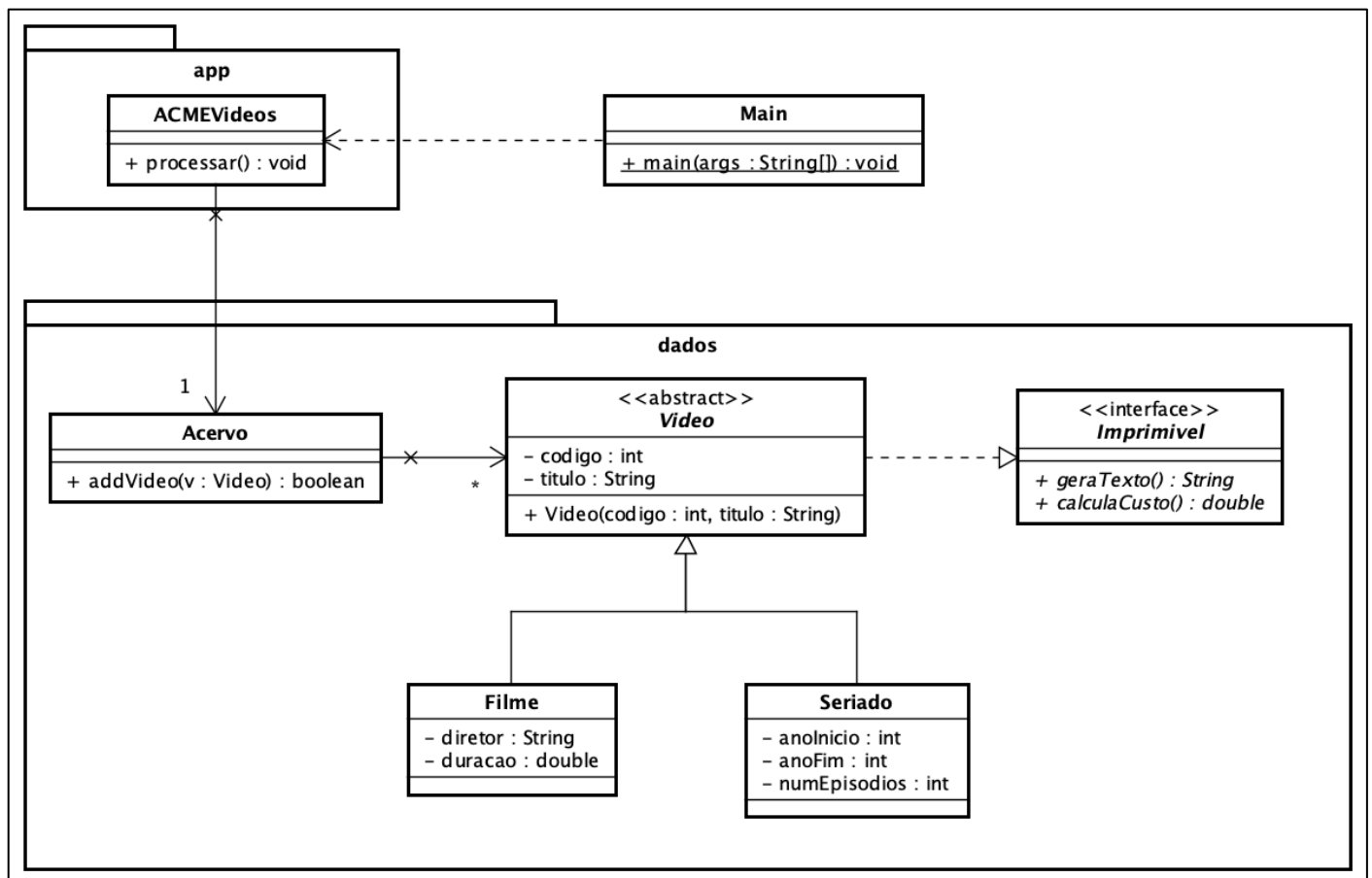
Exercício de Avaliação 2

1. Enunciado geral:

A ACMEVideos aluga vídeos (filmes e seriados) e deseja uma aplicação que faça o processamento de dados de seus vídeos.

Você será responsável pelo desenvolvimento da aplicação.

O analista de sistemas gerou um diagrama de classes inicial, com alguns atributos, operações e relacionamentos apresentados a seguir.



O analista definiu as seguintes operações da interface **Imprimivel**:

- **geraTexto()**: gera uma String a partir das informações do objeto, usa como separador de campos '- '.
- **calculaCusto()**: calcula o custo em Reais do objeto.

A classe **Acervo** é uma classe-catálogo que armazena a coleção de vídeos.

Sabe-se que será necessário haver subclasses da classe abstrata **Video**. Cada subclasse possui informações adicionais específicas:

- **Filme**: nome do diretor e o tempo de duração em minutos.
- **Seriado**: ano de início, ano de fim, número de episódios.

O método **calculaCusto()** depende da subclasse:

Subclasse	Valor da locação
Filme	R\$ 0,30 por minuto de duração
Seriado	R\$ 0,50 por episódio

O método **processar()** da classe ACMEVideos deve realizar a sequência de passos:

- lê do arquivo de dados de entrada todos os dados de cada vídeo e, se o código não for repetido no sistema, cadastra-o no sistema. Para cada vídeo cadastrado com sucesso no sistema, mostra os dados da vídeo no formato: **1: seguido do texto gerado pelo método geraTexto()**
Se o código do vídeo for repetido mostra a mensagem no formato: **1:Erro - código de video repetido.**
- Mostra os dados do vídeo com título mais longo, no formato: **2:código,título.**
Se não houver nenhum vídeo cadastrado mostra a mensagem no formato: **2:Erro - nenhum vídeo cadastrado.**
- Mostra os dados do vídeo com cálculo de custo mais baixo, no formato: **3:código,título,custo**
Se não houver nenhum vídeo cadastrado mostra a mensagem no formato: **3:Erro - nenhum video cadastrado.**
- Mostra os dados do seriado de maior período de exibição no formato: **4:código,título,período em anos**
Se não houver nenhum seriado cadastrado mostra a mensagem no formato: **4:Erro - nenhum seriado cadastrado.**
- Mostra o diretor com mais filmes no formato: **5:diretor,quantidade de filmes**
Se não houver nenhum filme cadastrado mostra a mensagem no formato: **5:Erro - nenhum filme cadastrado.**

2. Definição do exercício:

O objetivo do exercício é implementar um sistema que capaz de atender as necessidades da empresa descrita no enunciado geral, e que atenda as restrições a seguir:

- A entrada de dados ocorrerá por leitura de arquivo de texto.
- Os dados de entrada estarão no arquivo 'dadosentrada.txt': cada linha do arquivo possui um dígito no início. Se for 1 indica que é um Filme, se for 2 indica que é um Seriado.
- A saída de dados deve ser gravada no arquivo 'relatorio.txt'
- Informações fracionárias mostradas com 2 dígitos decimais.
- Toda entrada e saída de dados com o usuário deve ocorrer apenas na classe ACMEVideos.
- Para o armazenamento das mídias no sistema deve haver apenas uma lista de vídeos (List ou similar).
- Todos os atributos das classes devem ser privados.
- É permitida a criação de novos métodos, atributos e relacionamentos, mas as informações definidas no diagrama de classes original não podem ser alteradas.
- O diagrama de classes deve ser atualizado conforme as alterações realizadas e deve ser entregue em arquivo Astah ou PDF.

3. Critérios de avaliação

O exercício será avaliado conforme os seguintes critérios:

- Diagrama de classes atualizado: 1 ponto.
- Implementação de pacotes e interface: 1 ponto.
- Implementação e uso de generalização e polimorfismo: 2 pontos.
- Implementação correta conforme a descrição do exercício e o diagrama de classes: 3 pontos.
- Execução correta das opções previstas: 3 pontos.

- *Ponto extra (opcional) de 1 ponto (máximo de 10 pontos): implementar passo adicional:*
 - **Mostrar os dados do vídeo com menor desvio padrão do cálculo de custo:** calcula a média dos cálculos de custos dos vídeos cadastradas e localiza o vídeo com cálculo de custo mais próximo da média calculada, no formato: 6:média calculada, texto gerado pelo método `geraTexto()`

4. Entrega:

- A entrega do exercício envolverá:
 - arquivos dos códigos-fonte do sistema (e demais arquivos necessários para a compilação do sistema).
 - diagrama de classes atualizado.
- Deverá ser gerado um arquivo compactado (.zip ou .rar), com os itens acima, e entregue na tarefa da área Moodle da disciplina.
- Data de entrega: **4 / 11 / 2024**

5. Considerações finais:

- O exercício deve ser desenvolvido **individualmente**.
- A implementação deve seguir o Java Code Conventions para nomes de identificadores e estruturas das classes.
- Não será aceito exercício com erros de compilação. Programas que não compilarem corretamente terão nota zerada.
- A cópia parcial ou completa do exercício terá como consequência a atribuição de nota 0 (zero) aos exercícios dos alunos envolvidos. Para análise de similaridade será utilizado o MOSS (<https://theory.stanford.edu/~aiken/moss/>).