

Pontifícia Universidade Católica do Rio Grande do Sul  
Programação Orientada a Objetos  
Prof. Marcelo H. Yamaguti  
2024/1

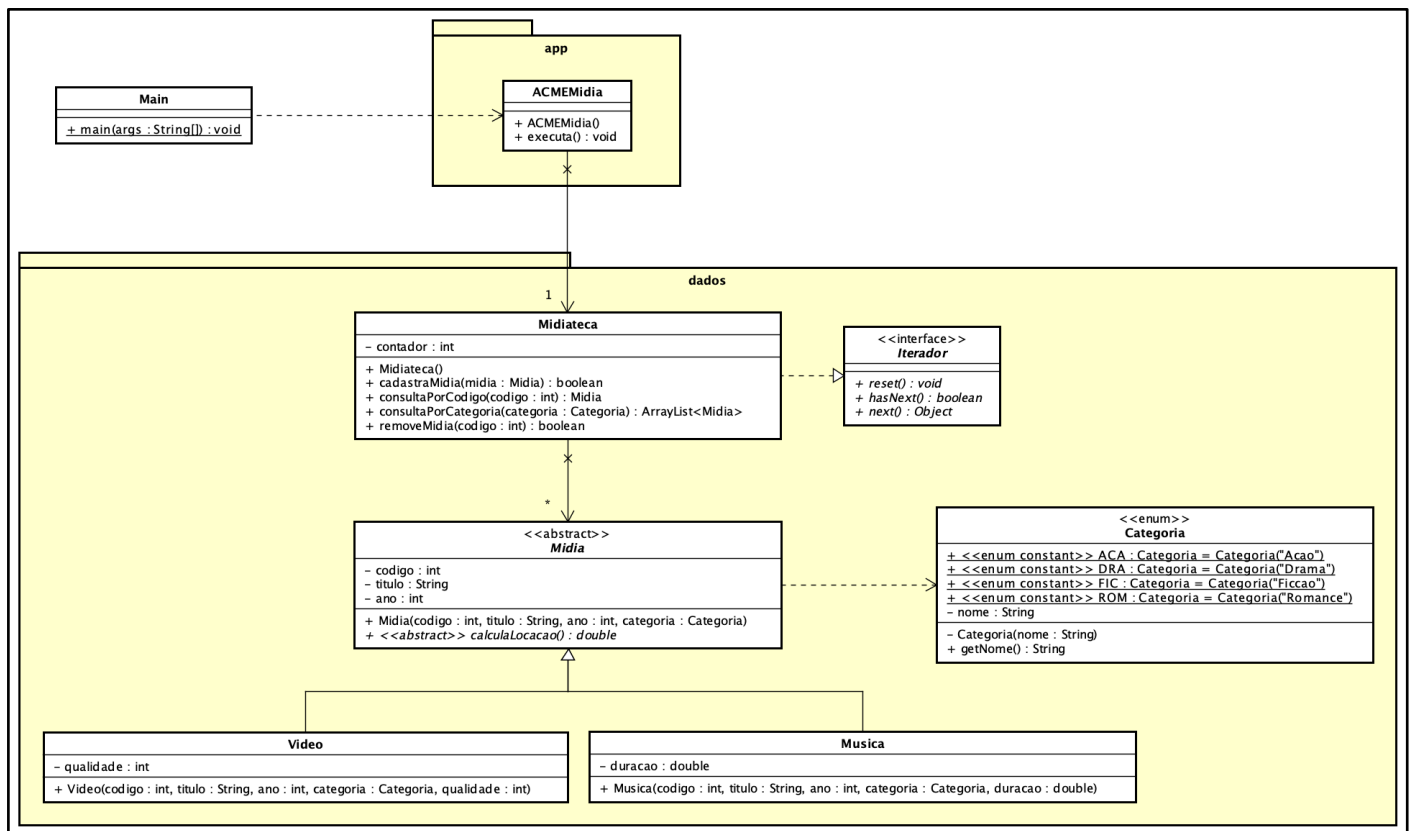
**Exercício de Avaliação 2**

**1. Enunciado geral:**

A ACMEMidia aluga mídias (vídeos e músicas) e deseja uma aplicação que faça o processamento de dados de suas mídias.

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 **Iterador**:

- **reset()**: reinicia a iteração na coleção.
- **hasNext()**: retorna *true* se ainda há elementos para a iteração, ou *false* em caso contrário.
- **next()**: retorna o próximo elemento da iteração.

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

- **Video**: possui a qualidade da resolução do vídeo.
- **Musica**: possui a duração em minutos.

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

- **Video**: considera a tabela a seguir:

| Ano  | Valor da locação |
|------|------------------|
| 2024 | R\$ 20,00        |

|               |           |
|---------------|-----------|
| 2000 a 2023   | R\$ 15,00 |
| Antes de 2000 | R\$ 10,00 |

- **Musica:** é calculado pela duração multiplicada com o valor por minuto da tabela a seguir:

| Categoria | Valor por minuto |
|-----------|------------------|
| Acao      | R\$ 0,90         |
| Drama     | R\$ 0,70         |
| Ficcao    | R\$ 0,50         |
| Romance   | R\$ 0,30         |

O método **executa()** da classe **ACMEMidia** deve realizar a sequência de passos:

1. **Cadastrar vídeos:** lê todos os dados de cada vídeo e, se o código não for repetido no sistema, cadastra-o no sistema. Se o código da vídeo for repetido mostra a mensagem no formato: **1:Erro-video com codigo repetido: codigo**  
Para cada vídeo cadastrado com sucesso no sistema, mostra os dados da vídeo no formato: **1:codigo,titulo,ano,categoria,qualidade**
2. **Cadastrar músicas:** lê todos os dados de cada música e, se o código não for repetido no sistema, cadastra-a no sistema. Se o código da música for repetido mostra a mensagem no formato: **2:Erro-musica com codigo repetido: codigo**.  
Para cada música cadastrada com sucesso no sistema, mostra os dados da música no formato: **2:codigo,titulo,ano,categoria,duração**
3. **Mostrar os dados de uma determinada mídia:** lê o código de uma mídia. Se não existir uma mídia com o código indicado, mostra a mensagem de erro: **3:Codigo inexistente**.  
Se existir, mostra os dados da mídia no formato: **3:atributo1,atributo2,atributo3,...,valor da locação**
4. **Mostrar os dados de mídia(s) de uma determinada categoria:** lê a categoria de uma mídia. Se não existir uma mídia com a categoria indicada, mostra a mensagem de erro: **4:Nenhuma midia encontrada**.  
Se existir, mostra os dados da(s) mídia(s) no formato: **4:atributo1,atributo2,atributo3,...,valor da locação**
5. **Mostrar os dados de vídeo(s) de uma determinada qualidade:** lê a qualidade de vídeo. Se não existir a qualidade indicada, mostra a mensagem de erro: **5:Qualidade inexistente**.  
Se existir, mostra os dados do(s) vídeos(s) no formato: **5:atributo1,atributo2,atributo3,...,valor da locação**
6. **Mostrar os dados da música de maior duração:** localiza a música cadastrada com maior duração. Se não existir nenhuma música cadastrada, mostra a mensagem de erro: **6:Nenhuma música encontrada**.  
Se existir, mostra os dados da música no formato: **6:titulo,duracao**
7. **Remover uma mídia:** lê o código de uma mídia. Se não existir uma mídia com o código indicado, mostra a mensagem de erro: **7:Codigo inexistente**.  
Se existir, mostra os dados da mídia no formato: **7:atributo1,atributo2,atributo3,...,valor da locação** e depois a remove do sistema.
8. **Mostrar o somatório de locações de todas as mídias:** calcula o somatório do valor de locação de todas as mídias do sistema. Se não existir mídia cadastrada no sistema, mostra a mensagem de erro: **8:Nenhuma midia encontrada**.  
Se existir, mostra a mensagem no formato: **8:valor do somatório**

## 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.
  - Pode-se utilizar de redirecionamento de E/S: ajuste a classe ACMEMidia para ler e escrever em arquivos: veja na área Moodle da disciplina > módulo: Materiais de apoio > CÓDIGOS AUXILIARES > Redirecionamento de entrada/saída de dados para arquivos.
  - Outra alternativa é a leitura e escrita em arquivos-texto.
- Os dados de entrada estarão no arquivo 'entrada.txt':
  - No passo **1. Cadastrar vídeos**: cada linha corresponde ao código, título, ano, categoria e qualidade de um vídeo. Quando a linha lida for -1, não há mais vídeos a serem cadastrados.
  - No passo **2. Cadastrar músicas**: cada linha corresponde ao código, título, ano, categoria e duração de uma música. Quando a linha lida for -1, não há mais músicas a serem cadastradas.
  - As últimas linhas do arquivo 'dadosin.txt' correspondem a:
    - Código da mídia para o passo 3.
    - Categoria da mídia para o passo 4.
    - Qualidade do vídeo para o passo 5.
    - Código da mídia para o passo 7.
- A saída de dados deve ser gravada no arquivo 'saida.txt'
- Toda entrada e saída de dados com o usuário deve ocorrer apenas na classe ACMEMidia.
- Para o armazenamento das mídias no sistema deve haver apenas uma lista de mídias (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 enumeração: 1 ponto.
- Implementação e uso de generalização e interface: 2 pontos.
- Uso de polimorfismo: 1 ponto.
- Implementação correta conforme a descrição do exercício e o diagrama de classes: 2 pontos.
- Execução correta das opções previstas: 3 pontos.
- *Ponto extra (opcional) de 1 ponto (máximo de 10 pontos): implementar passos adicionais:*
  - **Mostrar os dados da música com valor de locação mais próximo da média dos valores de locação**: calcula a média dos valores de locações das músicas cadastradas e localiza a música com valor de locação mais próximo da média calculada. Se não existir nenhuma música cadastrada, mostra a mensagem de erro: **9:Nenhuma musica encontrada**.  
Se existir, mostra os dados da música no formato: **9:média dos valores de locação,atributo1,atributo2,atributo3,...,valor da locação**
  - **Mostrar os dados da mídia mais nova**: mostra os dados da mídia mais nova. Se não existir nenhuma mídia cadastrada, mostra a mensagem de erro: **10:Nenhuma midia encontrada**.  
Se existir, mostra os dados da mídia no formato: **10:codigo,titulo,ano**

#### 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: ~~5 / 6 / 2024~~ **12 / 6 / 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/>).