

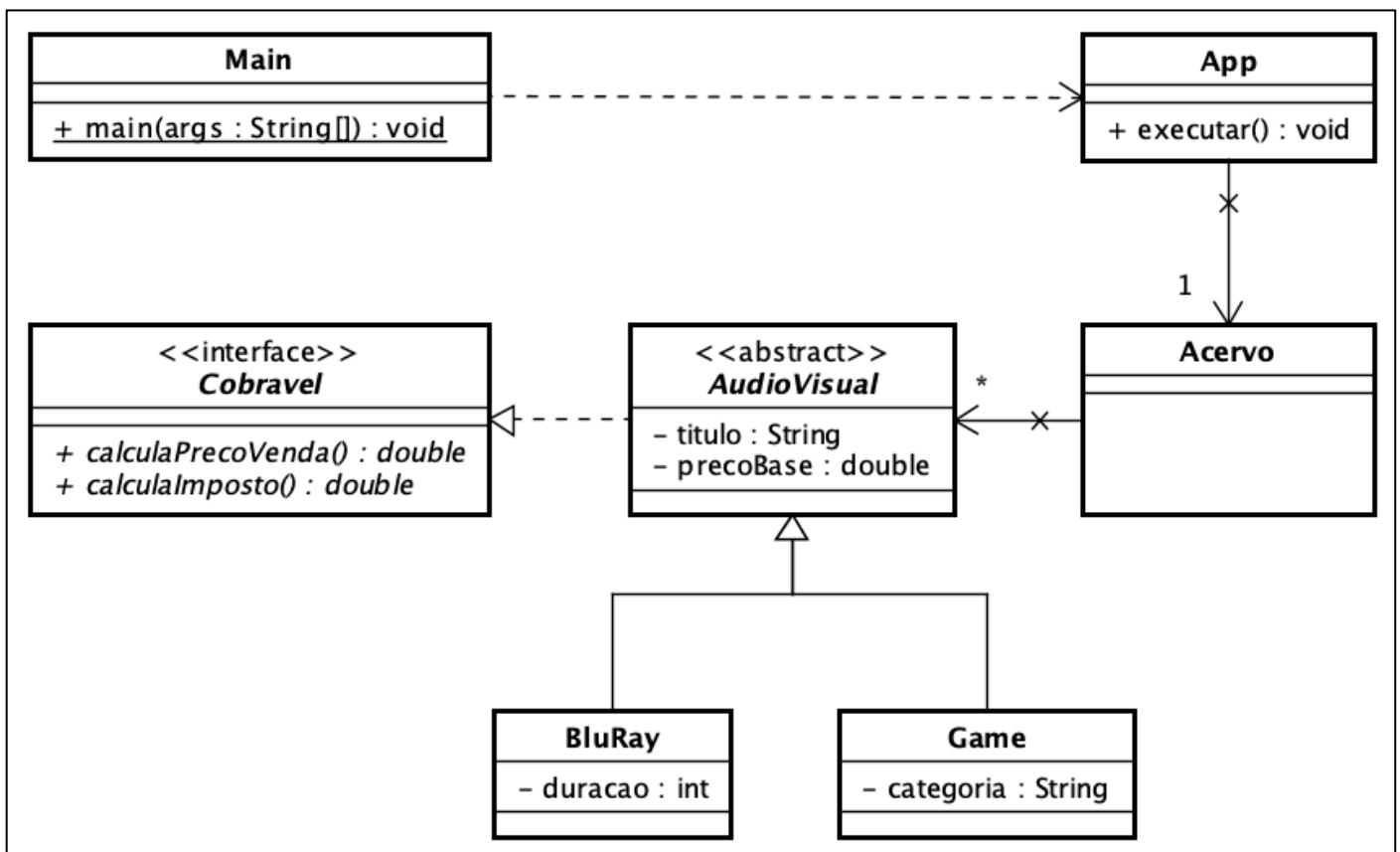
Exercício de Avaliação 2

1. Enunciado geral:

A ACMEFun comercializa itens de audiovisual (blu-rays e *games*) e já possui dados de seu acervo armazenados em arquivo-texto e deseja uma aplicação que faça o processamento de dados deste acervo.

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 **Cobravel**:

- **calculaPrecoVenda()**: calcula e retorna o preço de venda de um item.
- **calculaImposto()**: calcula e retorna o valor de imposto de um item.

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

- **BluRay**: possui uma duração (em minutos).
- **Game**: possui uma categoria (que pode ser: ACAO, ESPORTE, ESTRATEGIA, SIMULACAO ou RPG).

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

- **BluRay**: $\frac{\text{preço base} \times \text{duração}}{100}$

- **Game:** é o preço base acrescido de um percentual sobre o preço base conforme a categoria: 20% se for ACAO; 30% se for ESPORTE; 40% se for ESTRATEGIA; 50% se for SIMULACAO ou 70% se for RPG.

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

- **BluRay:** 40% do valor do preço de venda.
- **Game:** 50% do valor do preço de venda.

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

1. **Ler audiovisuais:** lê todos os dados de audiovisuais e escreve a quantidade de itens carregados com sucesso no formato: **1;quantidade de itens carregados**
2. **Escrever informações dos itens:** escreve algumas informações para cada item carregado com sucesso no sistema, no formato: **2;título;valor do preço final;valor do imposto**
3. **Escrever quantos games RPG foram cadastrados:** escreve a quantidade de games da categoria RPG que foram carregados no sistema no formato: **3;quantidade de games RPG**
4. **Escrever o BluRay com imposto mais próximo da média:** calcula a média dos valores de imposto de BluRays carregados e escreve qual BluRay possui o valor de imposto mais próximo da média calculada, no formato: **4:média dos valores de impostos de BluRays;título do BluRay com imposto mais próximo da média**. Caso não haja BluRays, escreve no formato: **4:Nenhum BluRay**

A leitura de dados deverá ser realizada no arquivo **dados.csv** e a escrita de dados deverá ser realizada no arquivo **resultado.csv**. Exemplos destes arquivos encontram-se no Apêndice.

2. Definição do exercício:

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

- É permitida a criação de novas classes, atributos e métodos; mas os relacionamentos, atributos e métodos definidos no diagrama de classes original não podem ser alterados.
- Toda entrada e saída de dados deve ocorrer nas classes App ou Acervo por meio de arquivos.
- Deve haver tratamento de exceções para que a aplicação não falhe na execução. Em caso de falha, deve ser apresentada no terminal (console) uma mensagem de erro com descrição da falha.
- 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.
- Uso de generalização e interface: 2 pontos.
- Uso de polimorfismo: 1 ponto.
- Tratamento de exceções: 1 ponto.
- Leitura e gravação de dados em arquivos-texto: 1 ponto
- Implementação correta conforme a descrição do exercício e o diagrama de classes: 2 pontos.
- Execução correta das opções previstas: 2 pontos.
- *Ponto extra (opcional): implementar e utilizar uma enumeração para o atributo 'categoria' da subclasse Game: 1 ponto (máximo de 10 pontos).*

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: **17 / 5 / 2023**

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/>).

Apêndice:

- **Estrutura de arquivo para entrada de dados (dados.csv):**

```
titulo;precobase;tipo;duração_categoria
```

- titulo: nome do audiovisual
- precobase: preço base do audiovisual
- tipo: 1 para BluRay, ou 2 para Game
- duracao_categoria:
 - se o tipo for 1, indica a duração em minutos do BluRay
 - se o tipo for 2, indica a categoria do game

- **Exemplo de arquivo para entrada de dados (dados.csv):**

```
Star Wars;11.11;1;121
Star Wars: Battle Front;131.11;2;ACAO
Dungeons and Dragons;222.22;2;RPG
Matrix;100.00;1;136
FIFA 23;333.33;2;ESPORTE
World of Warcraft;444.44;2;RPG
Gravidade;66.66;1;91
Xadrez;30.50;2;ESTRATEGIA
The SIMS;50.00;2;SIMULACAO
```

- **Exemplo de arquivo de saída de dados (resultado.csv):**

```
1;9
2;Star Wars;13.44;5.38
2;Star Wars: Battle Front;157.33;78.67
2;Dungeons and Dragons;377.77;188.89
2;Matrix;136.00;54.40
2;FIFA 23;433.33;216.66
2;World of Warcraft;755.55;377.77
2;Gravidade;60.66;24.26
2;Xadrez;42.70;21.35
2;The SIMS;75.00;37.50
3;2
4;28.01;Gravidade
```