

CATÁLOGO DE MÍDIAS

PROGRAMAÇÃO ORIENTADA A OBJETOS II

Data de entrega: até 25 de outubro de 2015.

Professor: Delano Medeiros Beder

1 Introdução

Um amigo seu que possui em casa quase 300 mídias (CDs ou DVDs) onde estão armazenados músicas, filmes e jogos, cansado de nunca encontrar os seus CDs/DVDs, fica sabendo que você está estudando Java e suplica a você que crie um programa para administrar as suas mídias.

Para quebrar o galho deste seu amigo, neste Exercício-Programa (EP), vamos implementar um catálogo de mídias em Java. A implementação será feita em duas partes: (1) escrita de classes para administrar uma coleção de mídias e (2) teste das classes implementadas.

1.1 Primeira Parte - Classes do Catálogo

O catálogo de mídias será implementado através das seguintes classes: **Catálogo**, **Mídia**, **CD**, **DVD** e **Jogo**. A classe **Catálogo** implementa a interface **ColeçãoMídia** e possui como atributo um *array* de tamanho máximo igual a 300. A classe **Catálogo** deve ter um construtor que recebe um parâmetro definindo o tamanho do *array*. Cada posição desde *array* poderá conter um objeto do tipo **Mídia** (As mídias são categorizados em: CDs de música, filmes em DVD e jogos).

Obs: Todas as classes e interfaces devem obrigatoriamente pertencer ao pacote **br.ufscar.si.catalogo**.

Classe Mídia

A classe abstrata **Mídia** deverá conter atributos para armazenar os seguintes dados sobre as mídias: título e ano de criação. Esta classe deverá também conter um ou mais métodos/construtores que permitam a atribuição de valores para cada um destes atributos.

A classe **Mídia** deve conter pelo menos os seguintes métodos/construtores:

- **Mídia(String título, int anoCriação)**. Construtor da classe que recebe os valores iniciais dos atributos título e ano de criação.
- **int getTipo()**. Método *abstrato* que deve ser implementado pelas subclasses de **Mídia**. Classe **CD** retorna 1, classe **DVD** retorna 2 e classe **Jogo** retorna 3.

- `void imprimeFicha()`. Método que imprime uma ficha contendo os dados da mídia semelhante às seguintes:

(a) Ficha de um CD de música

Título: Bachianas Brasileiras No.2
Ano: 2004
Tipo: CD de música
Artista: Orquestra de Câmara da Universidade de São Paulo
Faixa 1: (Prelúdio) O Canto do Capadócio, duração: 8:32
Faixa 2: (Ária) O Canto da Nossa Terra, duração: 6:29
Faixa 3: (Dança) Lembrança do Sertão, duração: 5:24
Faixa 4: (Tocata) O Trenzinho do Caipira, duração: 4:44

(b) Ficha de um filme em DVD

Título: O Senhor dos Anéis - A Sociedade dos Anéis
Ano: 2001
Tipo: Filme em DVD
Diretor: Peter Jackson
Artista 1: Elijah Wood, papel: Frodo Baggins
Artista 2: Viggo Mortensen, papel: Aragorn
Artista 3: Orlando Bloom, papel: Legolas Greenleaf
Artista 4: Christopher Lee, papel: Saruman
Artista 5: Ian McKellen, papel: Gandalf

(c) Ficha de um jogo

Título: <i>Need For Speed - Underground II</i>
Ano: 2005
Tipo: Jogo Eletrônico
Gênero: Corrida

Classe CD

A classe CD representa um CD de música e deve conter os seguintes atributos: artista (compositor/intérprete da obra), a lista de faixas (máximo 15) e duração de cada faixa em segundos. Essa classe deve conter pelo menos os seguintes métodos:

- `CD(String título, int anoCriação, String artista)`. Construtor da classe que recebe os valores iniciais dos atributos título, ano de criação e artista.
- `void adicionaFaixa(String faixa, int duração)`. Método usado para adicionar uma faixa (e sua duração) ao CD de música.
- `int getTipo()`. Reimplementação do método que retorna um inteiro que representa o tipo.
[CD = 1, DVD = 2 e Jogo = 3]
- `void imprimeFicha()`. Reimplementação do método que imprime uma ficha contendo os dados do CD.
[Exemplo: ver ficha (a) apresentada anteriormente]

Classe DVD

A classe DVD representa um filme em DVD e deve conter os seguintes atributos: diretor do filme e uma lista dos principais (máximo 5) atores/artistas que atuaram no filme e o papel desempenhado no filme. Essa classe deve conter pelo menos os seguintes métodos:

- DVD(String título, int anoCriação, String diretor). Construtor da classe que recebe os valores iniciais dos atributos título, ano de criação e diretor.
- void adicionaArtista(String artista, String papel). Método usado para adicionar um ator/atriz e seu papel à lista dos principais (máximo 5) atores/artistas que atuaram no filme.
- int getTipo(). Reimplementação do método que retorna um inteiro que representa o tipo.
[CD = 1, DVD = 2 e Jogo = 3]
- void imprimeFicha(). Reimplementação do método que imprime uma ficha contendo os dados do DVD.
[Exemplo: ver ficha (b) apresentada anteriormente]

Classe Jogo

A classe Jogo representa um jogo eletrônico e deve conter o seguinte atributo: gênero do jogo eletrônico (Esportes, Corrida, RPG, Aventura, Tabuleiro, etc). Essa classe deve conter pelo menos os seguintes métodos:

- Jogo(String título, int anoCriação, String gênero). Construtor da classe que recebe os valores iniciais dos atributos título, ano de criação e gênero.
- int getTipo(). Reimplementação do método que retorna um inteiro que representa o tipo.
[CD = 1, DVD = 2 e Jogo = 3]
- void imprimeFicha (). Reimplementação do método que imprime uma ficha contendo os dados do Jogo.
[Exemplo: ver ficha (c) apresentada anteriormente]

Interface ColeçãoMídia

A interface ColeçãoMídia define os seguintes métodos:

- Mídia[] coleção(). Retorna as mídias do catálogo.
- Mídia[] coleçãoPorTipo(int tipo). Retorna as mídias de um determinado tipo (CD, DVD ou Jogo).
[CD = 1, DVD = 2 e Jogo = 3]

Classe Catálogo

A classe Catálogo deve implementar a interface ColeçãoMídia e conter pelo menos os seguintes métodos:

- Catálogo(int tamMax). Construtor da classe que recebe o tamanho máximo do *array*. Cada posição deste *array* poderá conter um objeto do tipo Mídia.

- `boolean adicionaMídia(Mídia mídia)`. Adiciona uma mídia ao catálogo e devolve `true` se a inserção foi realizada com sucesso. A adição deve ser feita até adicionar o número máximo de mídias, caso no qual não deve ser feita nenhuma adição e o método deve devolver `false`.
- `Mídia obtêmMídia(String título)`. Método que retorna a mídia cujo título é igual ao parâmetro passado e `null`, caso a mídia não seja encontrada no catálogo.
- `int quantidadeMáximaDeMídias()`. Fornece a quantidade máxima de mídias que este catálogo pode armazenar.
- `int quantidadeDeMídias()`. Fornece a quantidade de mídias no catálogo.
- `int quantidadeDeCDs()`. Fornece a quantidade de CDs no catálogo.
- `int quantidadeDeDVDs()`. Fornece a quantidade de DVDs no catálogo.
- `int quantidadeDeJogos()`. Fornece a quantidade de jogos no catálogo.
- `Mídia[] coleção()` [definido pela interface `ColeçãoMídia`]. Retorna as mídias do catálogo.
- `void imprimeColeção()`. Imprime as fichas das mídias retornada pelo método anterior.
- `Mídia[] coleçãoPorTipo(int tipo)` [definido pela interface `ColeçãoMídia`]. Retorna as mídias de um determinado tipo (CD, DVD ou Jogo).
[CD = 1, DVD = 2 e Jogo = 3]
- `void imprimeColeçãoPorTipo(int tipo)`. Imprime as fichas das mídias retornada pelo método anterior.

1.2 Segunda Parte – Classe de Testes

Para ajudar na determinação do funcionamento de seu programa, será disponibilizada uma classe de testes `TestaCatálogo` com o método `realizaTestes`. Este método realiza diversas operações com os métodos de `Catálogo` para verificar se estes estão funcionando corretamente.

Atenção: O fato de sua classe funcionar para estes testes não significa que ela está totalmente correta. O tutor realizará outros testes diferentes daqueles da classe de teste fornecida.

2 Observações importantes

2.1 Sobre a elaboração:

- Este exercício-programa deve ser elaborado individualmente.
- Você deve utilizar **apenas** os conceitos apresentados em aula.
 - Você deve utilizar os **mecanismos de herança e interface** para a implementação do catálogo.
 - Os atributos das classes devem ser **privados**.
- É obrigatório que a classe de testes funcione corretamente com o seu EP!

- Compacte o código-fonte de todas as classes e interfaces em um arquivo `Catalogo<RA>.zip`
Exemplo: `Catalogo1234567.zip` (Cuidado para não enviar o arquivo errado!)
- Não é necessário compactar os arquivos compilados (.class).
- Guarde uma cópia do seu programa entregue.

2.2 Sobre a avaliação:

- Não serão toleradas cópias! Exercícios copiados (com ou sem eventuais disfarces) receberão nota ZERO. O exercício do aluno alvo da cópia também receberá nota ZERO.
- Exercícios com erros de sintaxe (ou seja, erros de compilação) receberão nota ZERO.
- A implementação das classes será avaliada segundo os seguintes critérios:
 - Soma simples dos valores obtidos nos itens de 1 a 2
 1. Atendimento às normas de boas práticas de programação (comentários, endentação, nomes de variáveis, estruturação do código, etc) [0..20]
 2. Corretude na implementação da atividade [0..80]
- É muito importante que seu programa tenha comentários e esteja bem endentado, ou seja, digitado de maneira a ressaltar a estrutura de subordinação dos comandos do programa (conforme visto em aula). A qualidade do seu trabalho sob esse ponto de vista influenciará sua nota!
- As informações impressas pelo seu programa na tela devem aparecer da forma mais clara possível. Este aspecto também será levado em consideração no cálculo da sua nota.
- Uma regra básica é a seguinte: do ponto de vista do tutor responsável pela correção dos trabalhos, quanto mais convenientemente apresentado estiver o seu programa, melhor será a disposição dele para dar-lhe uma nota generosa.

2.3 Sobre a entrega:

- O prazo de entrega é o dia 25 de outubro de 2015 às 23h55.
- A entrega será feita unicamente pelo ambiente moodle. Não serão aceitos trabalhos enviados por email.
- Compacte todas as classes e interfaces em um arquivo `Catalogo<RA>.zip` e entregue somente este arquivo no ambiente moodle (Cuidado para não enviar o arquivo errado!)