

ACH2002 – Introdução à Ciência da Computação II
EACH – SEGUNDO SEMESTRE DE 2008

Primeiro Exercício-Programa

Data de entrega: até 15 de setembro de 2008.

Professor: Delano Medeiros Beder

CATÁLOGO DE MÍDIAS

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 cursando ACH2002 e suplica a você que crie um programa para administrar as suas mídias.

Para quebrar o galho deste seu amigo, neste Exercício-Programa, vamos implementar um catálogo de mídias em Java. A implementação será feita em três partes: (1) escrita de classes para administrar uma coleção de mídias, (2) teste das classes implementadas e (3) extensão da classe do catálogo para que os seus dados possam ser gravados e lidos de arquivos em disco.

1.1 Primeira Parte - Classes do Catálogo

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

Classe Midia

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

- *Midia* (String titulo, int anoCriacao). 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 *Midia*.
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, duracao: 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/interprete 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 titulo, int anoCriacao, String artista)`. Construtor da classe que recebe os valores iniciais dos atributos título, ano de criação e artista.
- `void adicionaFaixa (String faixa, int duracao)`. 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 titulo, int anoCriacao, 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 titulo, int anoCriacao, String genero). 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]

Classe Catalogo

A classe Catalogo deve conter pelo menos os seguintes métodos.

- Catalogo (int tamMax). Construtor da classe que recebe o tamanho máximo do array. Cada posição desde array poderá conter um objeto do tipo Midia.
- boolean adicionaMidia (Midia midia). 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.
- Midia obtemMidia(String titulo). Método que retorna a mídia cujo título é igual ao parâmetro passado e **null**, caso a mídia não seja encontrada no catálogo.
- int quantidadeMaximaDeMidias (). Fornece a quantidade máxima de mídias que este catálogo pode armazenar.

- `int quantidadeDeMidias ()`. 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.
- `Midia[] lista ()`. Retorna as mídias do catálogo, ordenadas pelo título.
- `void imprimeLista ()`. Imprime as fichas das mídias retornada pelo método anterior.
- `Midia[] listaPorTipo (int tipo)`. Retorna as mídias de um determinado tipo (CD, DVD ou Jogo), ordenadas pelo ano + título. Isto é, as mídias são ordenadas levando em conta primeiro o ano. Caso duas mídias tenham o mesmo ano, então são ordenadas pelo título.
[CD = 1, DVD = 2 e Jogo = 3]
- `void imprimeListaPorTipo (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 `TestaCatalogo` com o método `realizaTestes`. Este método realiza diversas operações com os métodos de `Catalogo` para verificar se estes estão funcionando corretamente.

Atenção: O fato de sua classe funcionar para estes testes não significa que ela está totalmente correta, apenas que para os testes realizados ela funciona. Os monitores realizarão testes diferentes daqueles da classe de teste fornecida.

1.3 Terceira Parte – Serialização

Agora que o catálogo foi implementado e testado, só falta melhorarmos o programa um pouquinho mais para que o catálogo possa ser gravado em um arquivo em disco e posteriormente lido deste arquivo. Para implementar isso, utilizaremos um mecanismo poderoso da linguagem Java chamado de *seriação de objetos* (*object serialization*). Através deste mecanismo, é possível transformar um objeto em uma série de bytes, armazenando-os em um arquivo. Posteriormente, é possível desserializar o objeto, ou seja, ler os bytes do arquivo e transformá-los em objetos na memória do computador.

Como este processo é um pouco complicado, para facilitar vida de vocês, criamos a classe `SeriadorDeCatalogo` que está disponível no arquivo `SeriadorDeCatalogo.java` (não é preciso entender o código). Para utilizar essa classe, copie o arquivo `.java` para o diretório onde estão os seus `.java` (e compile). O uso da classe é bem simples, ela contém apenas os dois seguintes métodos.

```

/**
 * Grava um objeto da classe 'Catalogo' em um arquivo
 *
 * Parametros:
 * - catalogo - O 'Catalogo' que sera gravado no arquivo
 * - nomeDoArquivo - o nome do arquivo onde sera gravado o 'Catalogo'
 */
void gravaCatalogo(Catalogo catalogo, String nomeDoArquivo);

/**
 * Lê um 'Catalogo' de um arquivo
 *
 * Parametros:
 * - nomeDoArquivo - o nome do arquivo de onde sera lido o 'Catalogo'
 *
 * Valor de retorno: um objeto 'Catalogo' correspondente ao catalogo lido
 * do arquivo
 */
Catalogo carregaCatalogo (String nomeDoArquivo);

```

Veja a seguir um exemplo de uso, assumindo que a variável `catalogo` já armazena um catálogo com vários objetos de arte e também vários artistas.

```

> SeriadorDeCatalogo s = new SeriadorDeCatalogo();
> s.gravaCatalogo(catalogo, "catalogo.dat");
> Catalogo catalogo2 = s.carregaCatalogo("catalogo.dat");
> catalogo2.lista()

```

Só um pequeno detalhe: antes de usar a classe `SeriadorDeCatalogo` é necessário explicitar ao compilador Java que as classes do nosso catálogo serão seriadas. Para isso, basta inserir o texto `implements java.io.Serializable` depois do nome das classes. Quaisquer classes extras criadas para implementar o catálogo também devem implementar a interface `Serializable`.

Assim, o seu código deverá ser semelhante ao seguinte:

```

class Catalogo implements java.io.Serializable {
// blabla...
}
abstract class Midia implements java.io.Serializable {
// blabla...
}

```

Depois disso, crie o método `void testaSeriacao()` na classe `TestaCatalogo` que testa a serialização. Este método não precisa ser muito complicado, você pode, por exemplo, usar os métodos já implementados da classe `TestaCatalogo`.

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 interface e herança** para a implementação do catálogo e a marcação **JavaDoc** para documentação do programa.
 - Os atributos das classes devem ser **privados**.
- É obrigatório que a classes de testes funcione corretamente com o seu EP!
- Coloque todas as classes e interfaces em um arquivo Catalogo<nroUSP>.java
Exemplo: Catalogo1234567.java (Cuidado para não enviar o arquivo errado!)
- Não é necessário entregar: (i) os arquivos HTML gerados utilizando o JavaDoc e (ii) o arquivo SeriadorDeCatalogo.java.
- 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.
- É muito importante que seu programa tenha comentários e esteja bem indentado, 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 monitor 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 15 de setembro de 2008 às 24h.
- No início do arquivo, acrescente um cabeçalho bem informativo, como o seguinte:

```

/*****
/**  ACH 2002 - Introdução à Ciência da Computação II          **/
/**  EACH-USP - Primeiro Semestre de 2008                      **/
/**  <turma> - <nome do professor>                               **/
/**                                                              **/
/**  Primeiro Exercício-Programa                                **/
/**  Arquivo: <nome do arquivo>                                  **/
/**                                                              **/
/**  <nome do(a) aluno(a)>                                     <número USP> **/
/**                                                              **/
/**  <data de entrega>                                           **/
*****/
```

Não é obrigatório que o cabeçalho seja idêntico a esse, apenas que contenha pelo menos as mesmas informações.

- A entrega será feita unicamente pelo CoL. Não serão aceitos trabalhos enviados por email. Fiquem atentos, pois o CoL agora possui horário limite para entrega de trabalhos.
- Não é necessário entregar: (i) os arquivos HTML gerados utilizando o JavaDoc e (ii) o arquivo SeriadorDeCatalogo.java.
- Coloque todas as classes em um arquivo Catalogo<nusp>.java e entregue somente este arquivo pelo CoL. (Cuidado para não enviar o arquivo errado!)