DA1MCTA018-13SA - Programação Orientada a Objetos - Paulo Henrique Pisani - 2021.2

Descrição | Visualizar envios

[EP] Playlist com exceções

Data de entrega: segunda, 2 Ago 2021, 23:59
Arquivos requeridos: Playlist.java, ArquivoMultimidia.java, ArquivoVideo.java, ArquivoAudio.java (Baixar)
Tipo de trabalho: Trabalho individual

Playlist com exceções

Neste exercício, deve ser implementada uma classe chamada Playlist, que irá armazenar uma playlist de arquivos multimídia. Para isso, neste exercício, devem ser implementadas as classes públicas **Playlist**, **ArquivoMultimidia**, **ArquivoVideo** e **ArquivoAudio** (todas as classes devem estar no **pacote multimidia**). Além disso, também podem ser lançadas dois tipos de exceções e as classes para essas exceções devem ser implementadas também: **IndiceArquivoInvalido** e **QuantidadeLimiteArquivos**. A seguir são descritos detalhes sobre as situações em que essas exceções são lançadas e o que essas classes com as exceções devem conter.

Classe abstrata ***ArquivoMultimidia***:

- Construtor: public ArquivoMultimidia(String nomeArquivo, int tamanho) - construtor que inicializa o nome do arquivo e o tamanho.
- Métodos:
  - public final String getNomeArquivo() - retorna o nome do arquivo.
  - public final int getTamanhoArquivo() - retorna o tamanho do arquivo.
  - public final void setNomeArquivo(String nomeArquivo) - alterna o nome do arquivo.

Classe **ArquivoVideo** (subclasse de ArquivoMultimidia):

- Construtor: public ArquivoVideo(String nomeArquivo, int tamanho, int largura, int altura) - instancia um ArquivoVideo especificando o nome do arquivo, o tamanho e a resolução (largura e altura).
- Método: public String toString() - sobrescreve o método toString() de Object. O retorno do método deve ser no formato "Video: %s (%d)" (observe que não há acento), em que %s é o nome do arquivo e o inteiro indicado com %d é o tamanho do arquivo.

Classe **ArquivoAudio** (subclasse de ArquivoMultimidia):

- Construtor: public ArquivoAudio(String nomeArquivo, int tamanho, boolean audioHD) - instancia um ArquivoAudio especificando o nome do arquivo, o tamanho e se este arquivo tem áudio HD ou não.
- Método: public String toString() - sobrescreve o método toString() de Object. O retorno do método deve ser no formato "Audio: %s (%d)" (observe que não há acento), em que %s é o nome do arquivo e o inteiro indicado com %d é o tamanho do arquivo.

Classe **Playlist**:

- Construtor: esta classe deve possuir apenas o construtor sem parâmetros, que inicializa a playlist sem nenhum arquivo. Uma playlist pode ter no máximo 10 arquivos.
- Métodos:
  - **public void adicionarItem(ArquivoMultimidia arquivo)** - adiciona um arquivo mulmídia no final da lista de arquivos. Se, ao adicionar um arquivo, a playlist já estava com 10 arquivos (limite máximo), deve ser lançada a exceção QuantidadeLimiteArquivos e o arquivo não é adicionado. A exceção **QuantidadeLimiteArquivos** é subclasse de Exception (mas não de RuntimeException) e seu construtor deve chamar o construtor da superclasse Exception com a mensagem "Quantidade limite de arquivos foi atingida.".
  - **public void renomearItem(int indiceArquivo, String novoNomeArquivo)** - altera o nome de um arquivo na lista de arquivos da playlist. Se o índice especificado não for válido (não há arquivo no índice especificado), deve ser lançada a exceção IndiceArquivoInvalido. A exceção **IndiceArquivoInvalido** é subclasse de Exception (mas não de RuntimeException) e seu construtor deve chamar o construtor da superclasse Exception com a mensagem "Indice de arquivo invalido = %d" (observe que

não há acento na mensagem), em que o inteiro representado por %d é o valor do índice inválido que foi passado como argumento no método.

- **public void moverParaInicio(int indiceArquivo)** - move o arquivo especificado para o início da lista. Se o índice especificado não for válido (não há arquivo no índice especificado), deve ser lançada a exceção IndiceArquivoInvalido. A exceção IndiceArquivoInvalido é a mesma que o método renomearItem pode lançar e que foi descrita anteriormente.
- **public String[] listarArquivos()** - retorna um vetor de String com os retornos do método toString() dos arquivos na playlist. O comprimento desse vetor é a quantidade de arquivos adicionados na playlist.
- **public void ordenarArquivos(int tipo)** - ordena os arquivos da playlist de acordo com o tipo de ordenação especificado no parâmetro tipo.
  - *tipo=1* - Ordena os arquivos em ordem alfabética (lexicográfica) de nome. Caso a lista tenha mais de um arquivo com o mesmo nome, use a ordem crescente do tamanho do arquivo para desempate.
  - *tipo=2* - Ordena os arquivos em ordem crescente de tamanho do arquivo. Caso a lista tenha mais de um arquivo com o mesmo tamanho, use a ordem alfabética (lexicográfica) para desempate.

Dica: para comparar duas Strings, é possível usar o método compareTo da classe String. Por exemplo:

```
String str1 = "ABC";
String str2 = "DEF";
int comparacao = str1.compareTo(str2);
```

O retorno pode ser um valor negativo (quando str1 vem antes de str2 na ordem lexicográfica), positivo (quando str2 vem antes de str1 na ordem lexicográfica) ou zero (quando str1 é igual a str2). Mais detalhes em: https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/String.html#compareTo(java.lang.String)

Observação: Não é permitido o uso de funções de ordenação prontas (por exemplo, sort). Implemente um algoritmo de ordenação. Também não use o termo "sort" em nenhuma parte do código.

Importante: Submeta apenas as classes públicas descritas no enunciado (todas devem estar no **pacote multimidia**) e também as classes das duas exceções: IndiceArquivoInvalido e QuantidadeLimiteArquivos (as duas classes das exceções tem acesso package). O programa principal já existe no sistema de correção automática. As classes submetidas não podem realizar impressão de dados, utilizar import e java.util.

Casos de teste

Para realizar os testes, o programa de correção instancia um objeto Playlist (playlist1). Após isso, diversos métodos são chamados ao longo do teste. Formato dos casos de teste (que aparecem ao avaliar as classes no sistema de correção automática):

Entrada

- quantidade de arquivos a serem instanciados
- lista de arquivos a serem instanciados no formato [tipo] [nome do arquivo] [valores de atributos]
- métodos chamados

Saída

- verificação da classe ArquivoMultimidia e das classes das exceções QuantidadeLimiteArquivos e IndiceArquivoInvalido
- arquivos instanciados
- instanciação da classe Playlist
- métodos chamados

VPL

◄ [EP] Autenticação usuário

Seguir para... [EP] REC - Lâmpada Inteligente com exceções ►

(educadores/alunos) possam criar cursos, gerenciá-los e participar de maneira colaborativa.

Universidade Federal do ABC - Moodle (2020)

Obter o aplicativo para dispositivos móveis