

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

[Painel](#) / [Meus cursos](#) / [POO - DA1MCTA018-13SA - 2021.2](#) / [Construtores, cópia de objetos, final](#) / [\[EP\] Playlist](#)

Descrição

[Visualizar envios](#)

[EP] Playlist

Data de entrega: segunda, 12 Jul 2021, 23:59

Arquivos requeridos: Playlist.java, ArquivoMultimidia.java, ArquivoVideo.java, ArquivoAudio.java ([Baixar](#))

Tipo de trabalho: Trabalho individual

Playlist

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 **Playlist**, **ArquivoMultimidia**, **ArquivoVideo** e **ArquivoAudio** (todas as classes devem estar no **pacote multimedia**).

Classe abstrata **ArquivoMultimidia**:

- Construtor: public ArquivoMultimidia(String nomeArquivo) - construtor que inicializa o nome do arquivo
- Métodos:
 - public final String getNomeArquivo() - retorna o nome do arquivo
 - public final void setNomeArquivo(String nomeArquivo) - alterna o nome do arquivo

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

- Construtor: public ArquivoVideo(String nomeArquivo, int largura, int altura) - instancia um ArquivoVideo especificando o nome do arquivo 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 x %d)" (observe que não há acento), em que %s é o nome do arquivo e os dois inteiros indicados com %d são respectivamente largura e altura do vídeo.

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

- Construtor: public ArquivoAudio(String nomeArquivo, boolean audioHD) - instancia um ArquivoAudio especificando o nome do arquivo 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 (HD = %s)" (observe que não há acento), em que o primeiro %s é o nome do arquivo e o segundo %s pode assumir o valor "true" ou "false" (true se o arquivo tem áudio HD e false caso contrário).

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 100 arquivos.
- Métodos:
 - public void adicionarItem(ArquivoMultimidia arquivo) - adiciona um arquivo multimedia no final da lista de arquivos.
 - public void renomearItem(int indiceArquivo, String novoNomeArquivo) - altera o nome de um arquivo na lista de arquivos da playlist.
 - public void moverParaInicio(int indiceArquivo) - move o arquivo especificado para o início da lista.
 - public String[] listarArquivos() - retorna um vetor de String com os nomes dos arquivos na playlist. O comprimento desse vetor é a quantidade de arquivos adicionados na playlist.
 - public Playlist clone() - sobrescreve o método clone de Object. Esse método deve realizar uma **cópia profunda** do objeto Playlist. Nessa cópia profunda, inclusive as instâncias dos arquivos devem ser copiadas na nova instância retornada.

Importante: Submeta apenas as classes públicas descritas no enunciado (todas devem estar no **pacote multimedia**). O programa principal já existe no sistema de correção automática. As classes submetidas não podem realizar impressão de dados ou utilizar import.

Casos de teste

Para realizar os testes, o programa de correção instancia um objeto Playlist (playlist1). Ao longo do teste, o programa chama o método clone de playlist1 e assim obtêm uma instância que é armazenada em playlist2. Além disso, 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 ArquivoMultimedia
- arquivos instanciados
- instanciação da classe Playlist
- métodos chamados

[◀ \[EP\] Perfil de usuário](#)

Seguir para...

[\[EP\] Relatório com plugins ▶](#)

[VPL](#)



Este é o Ambiente Virtual de Aprendizagem da UFABC para apoio ao ensino presencial e semipresencial. Esta plataforma permite que os usuários (educadores/alunos) possam criar cursos, gerenciá-los e participar de maneira colaborativa.

Informação

- Conheça a UFABC
- Conheça o NTI
- Conheça o Netel

Contato

Av. dos Estados, 5001. Bairro Bangu - Santo André /SP – Brasil. CEP 09210-580.

Siga-nos



[Universidade Federal do ABC](#) - [Moodle](#) (2020)

[Obter o aplicativo para dispositivos móveis](#)