

Sumário

- Excepções

Documentação complementar Java

- [Exceptions](#)

Exercício 1

Recorrendo à linguagem Java, escreva o código necessário para dar suporte ao desenvolvimento de uma aplicação informática, chamada `PPodDemo`, que permita simular os populares *players* MP3 e tratar convenientemente situações de exceção.

O *player* a desenvolver é um `PPod`, limitado a 100 MB de espaço de memória, um máximo de 20 ficheiros, e capaz de reproduzir apenas ficheiros com extensão “mp3”. Os ficheiros têm as seguintes propriedades: nome, extensão, tamanho (em KB), e duração (em segundos).

O contrato da API: `PPod`, que permite dar suporte ao desenvolvimento da aplicação informática desejada, estipula as seguintes funcionalidades:

- `addFile(File file)` – adicionar um ficheiro ao *player*. Deve lidar com as exceções que ocorram ao longo do programa (*null references*, memória cheia, ou máximo de ficheiros atingido).
- `deleteFile(int index)` – apagar um ficheiro do *player*. Este método deve contemplar a exceção de índices inválidos. Esta operação só está disponível se existirem álbuns armazenados.
- `playTrack(int index)` – tenta reproduzir a faixa no índice indicado. A reprodução da faixa corresponde simplesmente a imprimir no ecrã os respectivos valores de “nome” e “duração”. Este método deve contemplar a exceção de índices inválidos ou para ficheiros com extensões não suportadas.
- `nextTrack()` – avança para a faixa seguinte, “reproduzindo-a” automaticamente. Caso não seja possível reproduzir o ficheiro corrente, deve saltar automaticamente para a faixa seguinte até encontrar um ficheiro passível de ser reproduzido.
- `previousTrack()` – recua para a faixa anterior, “reproduzindo-a” automaticamente. Caso não seja possível reproduzir o ficheiro na faixa corrente, deve saltar automaticamente para a faixa anterior até encontrar um ficheiro passível de ser reproduzido.

Implemente o contrato pedido.

Escreva o código necessário para testar o mais exaustivamente possível o código que desenvolveu como resposta ao que foi pedido. Sempre que for adequado, reutilize tipos de excepção da API Java. Quando os tipos de excepção da API Java não forem apropriados, declare um tipo de excepção específico desta aplicação.

A aplicação `PPodDemo` deve simular a execução do *player* de modo a provocar situações de falha e demonstrar a correcta reacção do `PPod` sem provocar o término extemporâneo do programa.

Exercício 2

Partindo do exercício 1, aumente a robustez e realismo do `PPod`, acrescentando as seguintes funcionalidades:

- O método `playTrack` gera exceção quando a duração do ficheiro é inválida.
- O método `addFile` gera exceção quando o tamanho do ficheiro é inválido.
- Um novo método `shufflePlay()` que reordena os ficheiros do player segundo um algoritmo com que o *player* foi configurado na inicialização e reproduz automaticamente a primeira faixa válida que encontrar, a partir do início da memória. A implementação do *player* não deve ter qualquer dependência sobre quaisquer algoritmos de reordenação. O *player* apenas sabe como invocar a reordenação. Implemente os seguintes algoritmos de reordenação: por nome (ascendente), por tamanho do ficheiro (descendente), e por duração da faixa (ascendente). A aplicação `PPodDemo` deve também testar *players* inicializados com diferentes algoritmos de reordenação.
- Implemente um método `list` que imprima no ecrã o nome das faixas válidas, ordenadas pelo índice.
- Implemente um método `getGlobalAddFileFailures` que permita à empresa produtora do `PPod` saber quantas vezes o método `addFile` originou exceções nos seus *players*. Note que esta informação é global a todos os *players* e não diz respeito a um *player* em particular.