

Aplicação de Geração de Gráficos de Graves a partir de Links do YouTube

Visão Geral

Esta aplicação Java demonstra como é possível extrair o áudio de um vídeo do YouTube, processar o sinal de áudio para identificar a intensidade dos graves e, em seguida, gerar uma representação gráfica desses dados. O projeto segue os princípios da Programação Orientada a Objetos (POO) para garantir modularidade, extensibilidade e facilidade de manutenção.

Arquitetura Orientada a Objetos

A aplicação é estruturada em torno de interfaces e classes que representam as diferentes responsabilidades do sistema:

- **IYoutubeAudioDownloader (Interface):** Define o contrato para qualquer classe que seja responsável por baixar o áudio de um link do YouTube. Isso permite que diferentes implementações de download (por exemplo, usando diferentes bibliotecas ou ferramentas) sejam facilmente trocadas sem afetar o restante da aplicação.
 - **YoutubeDownloader (Classe Concreta):** Uma implementação de `IYoutubeAudioDownloader` que utiliza a ferramenta de linha de comando `yt-dlp` para baixar o áudio do YouTube. Esta classe lida com a execução do processo externo e a extração do caminho do arquivo de áudio baixado.
- **AudioStream (Classe de Modelo):** Uma classe simples que representa o arquivo de áudio baixado. Atualmente, armazena apenas o caminho do arquivo, mas poderia ser estendida para incluir metadados do áudio ou o próprio fluxo de bytes.
- **IAudioProcessor (Interface):** Define o contrato para qualquer classe que seja responsável por processar o fluxo de áudio e extrair dados relevantes, como a intensidade dos graves.
 - **AudioSignalProcessor (Classe Concreta):** Uma implementação de `IAudioProcessor`. Atualmente, esta classe contém uma simulação do processamento de áudio e da extração de dados de graves. Em uma

implementação real, ela integraria uma biblioteca de processamento de sinal de áudio (como TarsosDSP) para realizar Análise de Fourier (FFT) e calcular a intensidade das frequências graves.

- **BassData (Classe de Modelo):** Uma classe simples que encapsula os dados de intensidade dos graves extraídos do áudio. Contém um array de `double s` que representam a intensidade dos graves ao longo do tempo.
- **IBassGraphGenerator (Interface):** Define o contrato para qualquer classe que seja responsável por gerar uma representação gráfica dos dados de graves.
 - **ConsoleBassGraphGenerator (Classe Concreta):** Uma implementação de `IBassGraphGenerator` que gera um gráfico de barras simples no console, usando asteriscos para representar a intensidade dos graves.
- **BassGraphApp (Classe Principal da Aplicação):** Esta é a classe que orquestra o fluxo de trabalho da aplicação. Ela recebe instâncias das implementações concretas das interfaces (`IYoutubeAudioDownloader` , `IAudioProcessor` , `IBassGraphGenerator`) através de seu construtor (Injeção de Dependência). Isso torna a aplicação flexível e fácil de testar, pois as dependências podem ser facilmente substituídas por mocks ou outras implementações.

Como Usar (Conceitual)

1. Pré-requisitos:

- Java Development Kit (JDK) 8 ou superior.
- Maven (para gerenciamento de dependências e construção do projeto).
- `yt-dlp` instalado e configurado em seu PATH (para o `YoutubeDownloader` funcionar).

2. **Construir o Projeto:** Navegue até o diretório raiz do projeto (`bass-graph-app`) no terminal e execute: `bash mvn clean install` Isso compilará o código e instalará as dependências.

3. **Executar a Aplicação:** Você pode executar a aplicação a partir da linha de comando: `bash java -jar target/bass-graph-app-1.0-SNAPSHOT.jar` Atualmente, o link do YouTube é hardcoded no método `main` para fins de demonstração. Você precisaria modificá-lo no código-fonte (`BassGraphApp.java`) para testar com diferentes vídeos.

Próximos Passos para uma Implementação Completa

- **Integração Real de Processamento de Áudio:** Substituir a simulação em `AudioSignalProcessor` por uma integração real com uma biblioteca como TarsosDSP. Isso envolveria:
 - Carregar o arquivo MP3 baixado.
 - Aplicar a Transformada Rápida de Fourier (FFT) para obter o espectro de frequência.
 - Analisar as bandas de frequência baixas para determinar a intensidade dos graves.
- **Visualização Gráfica Avançada:** Implementar uma nova classe `IBassGraphGenerator` que utilize uma biblioteca de UI Java (como JavaFX ou Swing) para criar um gráfico visualmente mais rico, em vez de apenas a saída do console.
- **Interface de Usuário:** Adicionar uma interface de usuário para permitir que o usuário insira o link do YouTube e visualize o gráfico de forma interativa.
- **Tratamento de Erros Robusto:** Melhorar o tratamento de erros e exceções em toda a aplicação.
- **Testes Unitários:** Escrever testes unitários para cada componente para garantir a corretude e facilitar futuras modificações.

Este projeto serve como um ponto de partida para a construção de uma aplicação mais completa e funcional.