AnubisGPT - Java Chatbot (Code Explanation)

This document explains the working mechanism of the `Anubius` Java chatbot, which uses Java Swing for the GUI, OpenRouter API for AI responses, and FreeTTS for voice output.

# 1. Libraries Used

* javax.swing.\* - for building the GUI interface.
* java.awt.\* - for layout and component styling.
* java.io.\* - for input and output stream handling.
* java.net.\* - for HTTP networking to call the API.
* java.util.ArrayList - to manage the list of messages.
* com.google.gson.\* - to parse JSON data (used for API request and response).
* com.sun.speech.freetts.\* - for text-to-speech functionality.

# 2. Core Functionalities

* User interface built using Java Swing with top bar, chat area, and input panel.
* Sends user messages to OpenRouter API and receives AI responses.
* Displays messages in a styled chat area.
* Reads out AI responses using FreeTTS when voice is enabled.

# 3. Key Components and Logic

## Constructor `Anubius()`

- Sets up the window, top bar with app name/logo.  
- Initializes FreeTTS and the voice engine.  
- Configures chat area, input panel, buttons (send, toggle voice).  
- Adds action listeners to send messages.

## `toggleVoice()`

Toggles voiceEnabled flag and updates the button label and color.

## `speak(String text)`

Speaks the text aloud if voice is enabled using FreeTTS.

## `showGreetingMessage()`

Displays an initial greeting message from the bot and speaks it.

## `sendMessage()`

- Takes user input from the text field.  
- Appends the user message to the chat area (right-aligned).  
- Calls `getAIResponse()` in a background thread to fetch AI response.  
- Appends the bot's response to the chat area and optionally speaks it.

## `getAIResponse(String prompt)`

- Connects to OpenRouter API using `HttpURLConnection`.  
- Constructs a JSON request body containing the model and user message.  
- Parses the JSON response to extract the AI's message.  
- Cleans the output by removing formatting symbols (\*, #, etc.).

# 4. API Mechanism

The app uses HTTP POST to communicate with OpenRouter's endpoint:  
URL: https://openrouter.ai/api/v1/chat/completions  
Headers:  
- Content-Type: application/json  
- Authorization: Bearer <API\_KEY>  
Body: JSON containing the model and message array.  
Response: JSON with AI-generated message in `choices[0].message.content`.

# 5. Entry Point

`main()` method launches the application using `SwingUtilities.invokeLater`.

# 6. Note

Make sure all dependencies (FreeTTS, GSON) are correctly added to the classpath.