📘 Java Project Report: Mood-Based Music Recommender App

📌 Objective:

To develop a Java-based console application that analyzes user mood based on natural language input and suggests appropriate music along with wellness tips. This project integrates Object-Oriented Programming (OOP) with basic Natural Language Processing (NLP) logic for emotion detection.

🧠 Project Summary:

The Mood-Based Music Recommender is a console application that asks the user to describe their current feelings in natural language. The system analyzes the text, determines the user's mood using keyword-based NLP logic, and recommends a music track and a self-care tip.

🔧 Technologies Used:

Component	Technology
Programming Language	Java
Concepts	OOP (Encapsulation, Inheritance, Abstraction, Polymorphism)
NLP	Keyword Matching
Input	Scanner (console)
Output	Console Text
Optional Future Add-ons	JavaFX GUI, Spotify API, Sentiment Analysis API

🧩 Features:

- Takes natural language input from the user.
- Detects mood using keyword analysis.
- Recommends a song based on the detected mood.
- Provides a tip or advice matching the emotional state.
- Demonstrates all four pillars of Java OOP.

🏗️ Class Structure:

1. `UserMood.java`

- Stores user input and detected mood.
- Implements **encapsulation** via private fields and public getters/setters.

2. `MoodAnalyzer.java` (Abstract Class)

- Declares abstract method `analyzeMood(String text)`.
- Demonstrates **abstraction**.

3. `SimpleMoodAnalyzer.java`

- Extends `MoodAnalyzer`.
- Implements `analyzeMood()` using keyword matching logic.
- Demonstrates **polymorphism** and **inheritance**.

4. `SongRecommender.java`

- Contains static methods to return a song and tip based on mood.
- Central suggestion engine of the app.

5. `Main.java`

- Handles user input and coordinates all class interactions.
- Controls the program workflow.

🧪 Sample Input & Output:

🔹 Input:

```
How are you feeling today?
> I'm a bit tired and low, just finished a lot of work.
```

🔹 Output:

```
🎯 Detected Mood: Sad 😔
🎶 Suggestion: "Let It Be - The Beatles"
✨ Tip: Take a 10-min walk or listen to soft lo-fi music.
```

🧠 Logic Explanation:

✅ Mood Detection (Simple NLP)

- User text is converted to lowercase and scanned for keywords.

- Keywords like `happy`, `excited` → **Happy**
- Keywords like `tired`, `low`, `sad` → **Sad**
- Keywords like `angry`, `annoyed` → **Angry**
- Keywords like `calm`, `peaceful` → **Calm**
- If no keywords match → **Neutral**

🧱 Java OOP Concepts Used:

Concept	Example in Code
Encapsulation	`UserMood` has private fields
Abstraction	`MoodAnalyzer` as abstract class
Inheritance	`SimpleMoodAnalyzer` extends `MoodAnalyzer`
Polymorphism	Different analyzers can override `analyzeMood()`

🔄 Future Enhancements:

- Replace keyword matching with ML sentiment analysis (e.g., OpenNLP or Stanford NLP).
- Connect to Spotify API for real music playback.
- Add GUI interface using JavaFX.
- Allow spoken mood input via microphone (Speech-to-Text API).
- Track mood history with file/database storage.

📌 Conclusion:

This project showcases how Java OOP concepts can be used to build intelligent, real-life applications. The system bridges emotion recognition with music recommendation, offering both functionality and empathy. It’s a unique beginner-level project demonstrating creativity, problem-solving, and software design.

🧑‍💻 Developed By:

Swapnil Mortale
#100DaysWithCode – Day 4 Project
Language: Java
Platform: Console (Text-based)