Implement a small service (either an API or a batch process) in Java that fetches weather information in Japan and provides it to the end user in a convenient format.

You are free to choose the implementation style (REST API, CLI utility, batch job, etc.).

Requirements

1. Data Source

For example, you can use any public weather API to retrieve current weather data for Japan  
(e.g., OpenWeatherMap, WeatherAPI, or others).

Minimum required data: temperature, humidity, and weather description (sunny, rainy, cloudy, etc.).

2. Functionality

The service must return weather data at least for the following cities:

Tokyo

Osaka

Kyoto

Output should be provided in a structured format (JSON / CSV / text — depending on your chosen approach).

Error handling should be implemented (e.g., when the API is unavailable or the response is invalid).

Technology Stack

Main language: Java (version 11+).

You may use third-party libraries for HTTP requests (e.g., HttpClient, OkHttp, Spring RestTemplate, etc.).

If building an API service — prefer Spring Boot.

If building a batch process — a simple console application with readable output is sufficient.

Code Quality

Code must be structured, readable, and follow basic Clean Code / SOLID principles.

Core logic must be covered with unit tests (JUnit, TestNG, or other frameworks).

Include a README.md file with:

instructions on how to run the project,

list of dependencies,

example usage (REST calls or console output).

Bonus (optional, but nice to have)

Docker support for containerization.

CI/CD pipeline (e.g., GitHub Actions) for running tests automatically.

Swagger/OpenAPI documentation if implemented as a REST API.

Deliverables

A GitHub/GitLab/Bitbucket repository with source code.

Unit tests included.

README.md with setup and usage instructions.