Java Implementation of the Satellite Command System

Description

This directory contains the Java implementation of the Satellite Command System. The system simulates controlling a satellite's orientation, solar panel status, and data collection.

Adherence to SOLID Principles and Evaluation Criteria

SOLID Principles

- **Single Responsibility Principle (SRP):** Each Java class follows the SRP, having a single responsibility. For example, the `Satellite` class manages satellite state, while action classes handle specific operations.
- **Open/Closed Principle (OCP):** The implementation is designed to be open for extension but closed for modification. You can add new actions without changing existing code.
- **Liskov Substitution Principle (LSP):** The action classes implement a common `SatelliteAction` interface, ensuring interchangeability without affecting behavior.
- **Interface Segregation Principle (ISP):** The `SatelliteAction` interface is minimal, ensuring that classes implementing it don't depend on unused methods.
- **Dependency Inversion Principle (DIP):** High-level modules depend on abstractions (like `SatelliteAction`), not concrete classes, promoting flexibility.

Logging in Java Implementation

Logging plays a crucial role in this Java implementation. The project uses Java's built-in `java.util.logging` framework for logging satellite operations and errors. Logs are configured via the `logging.properties` file for customization.

- **Logging Configuration (`logging.properties`):** The `logging.properties` file defines log handlers, formatters, and log levels. This allows for fine-tuning log output.

Evaluation Criteria

- **Code Quality:** The code follows Java best practices, including clear naming conventions, encapsulation, and modular design.
- **Functionality:** The Java implementation accurately simulates satellite operations and maintains state.
- **Global Convention:** Java coding conventions are followed for readability and maintainability.
- **Gold Standards:** Logging provides detailed insights into satellite operations and any errors encountered during execution.

Requirements

- Java JDK 11 or higher

- Maven (for building and running the application)

Installation and Execution

To compile and run the Java implementation, navigate to the `Java/` directory and use Maven commands:

```
mvn compile
mvn exec:java -Dexec.mainClass=com.yourcompany.satellite.Main
```

Features

- Implementation of satellite commands like rotation and solar panel control.
- Comprehensive logging of satellite operations and errors.
- Object-oriented design following SOLID principles.

Refer to the Java source files in `src/main/java/com/yourcompany/satellite` and its subdirectories for detailed implementation.