Project 1 - Bonus

We use a GitHub workflow to automate the detection and refactoring of design smells in a repository. This approach ensures consistent and timely identification of code issues, improving code quality and maintainability. The workflow is triggered on every push event to the master branch of the repository. It consists of the following steps:

Automated Design Smell Detection

1. Checkout Repository:
This step uses the *actions/checkout* action to clone the repository into the GitHub Actions runner environment, allowing access to the repository's files.

2. Clone the Utils Repository:
The script clones a separate repository (*se_project_utils*) that contains utilities required for the design smell detection and refactoring process.

3. Set up JDK:
This step configures the Java Development Kit (JDK) version 11 using the actions/setup-java action.

4. Compile and Run Java File:
It executes a Java JAR file (DesigniteJava.jar) from the cloned utils repository, a tool for detecting design smells in Java code.

Automated Refactoring

5. Set up Python:
The workflow sets up a Python environment using the actions/setup-python action to prepare for executing Python scripts.

6. Install Dependencies:
This step installs the required Python dependencies specified in the requirements.txt file from the cloned utils repository.

7. Run Python File:
The openai_refactor.py Python script from the utils repository is executed. This script interfaces with OpenAI APIs to refactor the identified design smells in the codebase. It:

1. Reads data from a CSV file containing information (given by designite) about detected design smells.
2. Iterates over each design smell, finding the affected Java code file and extracting its content.
3. Calls the OpenAI API to generate refactored code for the detected design smell.
4. Overwrites the original Java code file with the refactored version.

8. Remove Extra Files:
This step cleans up the environment by removing the cloned utils repository and any temporary files generated during the workflow.

Pull Request Generation

9. Create Pull Request:
Finally, the workflow uses the *peter-evans/create-pull-request* action to automatically create a pull request with the refactored code changes. It includes a commit message, title, and body to provide information about the automated refactoring process.