**Fitness Tracker Documentation**

**1. Clean Code Practices**

Clean code practices ensure that the application is readable, maintainable, and efficient. My code adheres to clean code principles:

**1.1 Meaningful Variable and Method Names**

* Variables and methods are named clearly to indicate their purpose.
* Example: logWorkout(Workout workout) clearly represents adding a workout to the tracker.

**Code Example:**  
**

**1.2 Proper Encapsulation and OOP Principles**

* The Workout class encapsulates properties like type, duration, caloriesBurned, and date.
* Getters and setters are used to enforce encapsulation and prevent direct access to instance variables.

**Code Example:**  
*A screenshot of a computer program

AI-generated content may be incorrect.*

A computer screen shot of a program code

AI-generated content may be incorrect.

**1.3 Unit Testing with JUnit 5**

* Each test case covers both positive and negative scenarios.
* The testGoalTracking() method ensures that the goal calculation logic functions correctly.

**Code Example:**  
*A screenshot of a computer

AI-generated content may be incorrect.*

A screenshot of a computer code

AI-generated content may be incorrect.

**2. Project Overview**

**2.1 Project Purpose**

The Fitness Tracker application allows users to log workouts, track their fitness goals, and check if they have met their weekly calorie-burning target.

**2.2 How It Works**

* **Logging Workouts:** Users can add new workouts by specifying the type, duration, calories burned, and date.
* **Tracking Progress:** The application calculates total calories burned and checks if the set goal is met.
* **Goal Setting:** Users can define a weekly fitness goal.

**2.3 Test Cases**

We implemented the following test cases:

1. **testLogWorkout()** - Ensures workouts are logged correctly.
2. **testWorkoutHistory()** - Checks if multiple workouts are stored properly.
3. **testGoalTracking()** - Verifies that the application correctly determines if the goal has been achieved.
4. **testGoalNotMet()** - Ensures the application correctly identifies when the goal is not yet reached.

**3. Dependencies**

The following dependencies are required for this project:

* **JUnit 5** for unit testing.
* **Maven** for dependency management.

**3.1 Where the Dependencies Were Sourced From**

* Dependencies were added via the Maven pom.xml file.
* JUnit 5 was imported from the Maven Central Repository.

**Dependency Example:**  
<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>org.example</groupId>  
 <artifactId>fitnessTracker</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <dependencies>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>RELEASE</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <properties>  
 <maven.compiler.source>17</maven.compiler.source>  
 <maven.compiler.target>17</maven.compiler.target>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
  
</project>

**4. Challenges Faced**

During the development of this project, I encountered the following issues:

**4.1 Test Failures Due to Incorrect Assertions**

* Initially, some tests failed because assertions were comparing objects incorrectly.
* Solution: Used assertEquals() for value comparisons and ensured correct data retrieval from methods.

**4.2 Handling Edge Cases in Goal Tracking**

* Issue: The isGoalMet() method didn’t account for workouts logged across multiple weeks.
* Solution: Updated the logic to consider only workouts within the current tracking period.

**Conclusion**

This project successfully implements a simple yet functional **Fitness Tracker** using Java. It follows clean coding practices, including unit tests with JUnit 5. Future improvements may include implementing a database for persistent storage and a graphical user interface (GUI) for better usability.