

Objective

The objective of this lab is to demonstrate how writing good unit tests helps detect bugs.

Instructions

1. Download the zip file provided and extract the code for the exercises. Code for the three exercises are in packages named ex1, ex2, and ex3 under src/main/java.
2. Set the JAVA_HOME environment variable to your JDK installation directory. Then run the gradlew script on Linux or gradlew.bat on Windows to install the Gradle build tool. Then run “gradle build” build the code.
3. Put your tests under src/test/java in packages ex1, ex2 and ex3. To download JUnit dependencies and run the unit tests do “gradle test” If you need details of failing tests add the “--info” flag to this command.

These instructions assume you are writing tests using JUnit version 5. If you experience problems with these steps please ask on the [Google Classroom forum](#). The class registration code is up75knb.

Exercises

The code provided for each of these exercises has various bugs in the locations marked with the comment //XXX. Write one or more unit test cases for each method that uncover these bugs. Your tests should provide informative failure messages describing what the bug is. DO NOT fix the bugs or change the code provided in any way!

1. The HistoricalData class stores double valued data and calculates the sum and maximum of the data.
2. The Clock class maintains the current time. The tick method advances the time by one second when called.
AlarmClock extends Clock with functionality to set an alarm. If the alarm time is set the alarm starts ringing when the set time is reached. When the next minute is reached the alarm must switch off.
3. The code for Exercise 3 implements the following scenario. Developers love ice cream and their favorite is named Maxibon. They start with 10 Maxibons in the office fridge. Every time a group of one or more developer goes to the fridge they grab some maxibons:
 - Pedro grabs three maxibons.
 - Davide does not grab any maxibon.
 - Alberto grabs one maxibon.

- Jorge grabs two maxibons.
- Sergio grabs one maxibon.

If there are fewer maxibons left than what the developer wants, he takes the maxibons left. Once there are 2 or fewer they need to order more by sending a message through Slack saying "Hi guys, I'm <DEVELOPER_NAME>. We need more maxibons!" The secretary then orders a pack of 10 more.