Problem #1: I am writing into classpath `output.Demo`. This class references code from the MySQL Java DataBase Connector (JDBC) 8.0.23, and OpenJFX SDK 11.0.2. I hand-compile everything using the Windows command prompt and custom batch files. I am unable to properly utilize classes from the JDBC's `mysql-connector-java-8.0.22.jar` file. Specifically, there is a Java runtime error when the program attempts to access any class from the `com.mysql.cj.xdevapi` classpath. This runtime error is a `java.lang.NoClassDefFoundError`. There is no compile-time error; the program, including references to this classpath, compiles correctly. There is only an error at runtime.

View the code and error here.

(dlmountain's github profile -> weird-jvm project -> episode_one -> errors -> demo -> Demo.java)

Problem #2: While unit-testing my code, I removed references to `javafx` classpaths. This induced a compile-time error where the JVM asked me to modify the `main` method to have parameters `main(String args[])`. This parameter is required for the successful utilization of OpenJFX SDK's `Application` class, and all subclasses thereof. Obviously, the JVM recognized my code as being a subclass of `Application`, despite my having removed all references to OpenJFX SDK from the code. Thomas Royall advised that I also remove all OpenJFX SDK files from the directory tree of the project, fearing the JVM might be discovering these classpaths by enumerating all files in the directory tree. Removing these files did not help.

View the code and error here.

(dlmountain's github profile -> weird-jvm project -> episode_one -> errors -> yeet -> Yeet.java)

Problem #3: While unit-testing my code, I created a new directory with a single `.java` file. I compiled this file and ran it using a batch file. This raised a runtime error of `java.lang.NoClassDefFoundError`.

View the code and error here.

(dlmountain's github profile -> weird-jvm project -> episode_one -> errors -> broken -> Broken.java)