1. See logger.log, why is it different from the log to console?

The logger is configured by reading the the logger.properties file which sets up logging to the console and to the file logger.log, this log takes all levels and uses SimpleFormatter. The console is configured to receive INFO level of data, while the log Is configured to receive ALL level of data.

//MENTION ALL VS INFO

1. Where does this line come from? FINER org.junit.jupiter.engine.execution.ConditionEvaluator logResult Evaluation of condition [org.junit.jupiter.engine.extension.DisabledCondition] resulted in: ConditionEvaluationResult [enabled = true, reason = '@Disabled is not present']

This log comes from the failed test and it reveals that the failed test is enabled and still runs, so it is not overridden with the command “Disabled”.

1. What does Assertions.assertThrows do?

It asserts that the specific exception to be thrown is a TimerException

1. See TimerException and there are 3 questions

1. What is serialVersionUID and why do we need it? (please read on Internet)

The serialVersionUID attribute is an identifier that is used to serialize/deserialize an object of a Serializable class. It ensures same class Serialized was deserialized. The UID is automatically-generated based on a class name, implemented interfaces, and all public and protected members. Changing any of these in any way will change the serialVersionUID.

2. Why do we need to override constructors?

This way we can use getMessage and getCause in order to understand why they were thrown and the information they were thrown with. This allows us to extend the TimerException so that it handles the data we need to record in the program.

3. Why we did not override other Exception methods?

Other exceptions are autogenerated and imported, this one is created by us. They are inherited by their parent, and we do not need any specialized messages from them.

1. The Timer.java has a static block static {}, what does it do? (determine when called by debugger)

The static block is done first when Timer is constructed, it configures the logger. It does not need to be reconfigured multiple times throughout the run.

1. What is README.md file format how is it related to bitbucket? (https://confluence.atlassian.com/bitbucketserver/markdown-syntax-guide-776639995.html)

If your repository contains a README.md file at the root level, Bitbucket displays its contents on the repository's Source page if the file has the .md extension. The file can contain Markdown and a restricted set of HTML tags. This specific readme is configured as an ordered list format that displays data in numbered bullets.

1. Why is the test failing? what do we need to change in Timer? (fix that all tests pass and describe the issue)

The logger is reporting a time that has not been initialized, so I initialized it to system time beforehand, so there was no null error. It was expecting a timer exception, but instead received a null pointer exception because of the uninitialized data. By fixing the initialization the correct exception can be thrown.

1. What is the actual issue here, what is the sequence of Exceptions and handlers (debug)

There is a timerException if the time is not valid, an interrupted exception, if the sleep is interrupted, and then prints the run time. The actual issue is that it throws the exception and continues to try to solve the problem. The real issue is that the time exception does not stop the run.

1. Make a printScreen of your eclipse JUnit5 plugin run (JUnit window at the bottom panel)

Text

Description automatically generated

1. Make a printScreen of your eclipse Maven test run, with console

Text

Description automatically generated

1. What category of Exceptions is TimerException and what is NullPointerException

NullPointException and TimerException is a runtime exception. These are not checked because there is no catch block to handle these errors.

1. Push the updated/fixed source code to your own repository.

https://github.com/smadal/Assignment10