Georgi Ivanov

Statistics tracker (Java)

Instructions

1. Overview
    - The project consists of three classes: RollingStatistics and RollingStatisticsTimestamped which are very similar. RollingStatistics is the center of the assignment and RollingStatisticsTimestamped is an implementation of the assignment extension that asks if the maximum range of data could be stored over a specified time rather than a specified amount. DataItem is the third class which is used to represent a piece of data in the timestamped version of the assignment.

2. Main method instructions
    - The Main class simply consists of a main method which can be used to dynamically test the application and see it in action.
    - Once you run the Main class as a java application you will be prompted from the console to select which way of storing data you would like to use. Option 1 is the storage with a specified maximum amount of data and option 2 is the storage over a specified time.
    - If you choose option one, you will be asked to provide the maximum amount of data your storage can hold. Having done that, the application will prompt you to enter a command. Your options are:
        - "add {number}" – this is going to add the number you provide to the storage. For example, "add 3.14" is going to add 3.14 to the data storage.
        - "statistics" – this is going to show you information about all the data you have stored, such as: Sum, Mean, Standard Deviation, etc.
        - "range {integer} {integer}" – this is going to show you information for a specified range of data in the storage. For example, if you have the items [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] in your storage. Running "range 2 5" is going to provide you with full information about the items [2, 3, 4, 5].
        - "back" – this takes you one step back in the running application.

    - If you choose option two, you will be asked to provide the amount of time (seconds) your data will be kept in storage, and the delay (seconds) in

between every check for old items in your storage. This check simply goes over the items in your storage and removes the ones that are “too old”. Having done that, the application will prompt you to enter a command. Your options are:
    - “add {number}” – this is going to add the number you provide to the storage. For example, “add 3.14” is going to add 3.14 to the data storage and its start time will be the time at which the command is executed.
    - “statistics” – this runs a check for “old data” on your storage and once again displays full information about all the items.
    - “range {integer}” - this runs a check for “old data” on your storage and displays full information about items that have been added in the last {number} seconds. For example: “range 30” will display information about all the items that have been added to the storage in the last 30 seconds.
    - “check” – this will manually perform a check for “old data” on your storage.
    - “back” – this takes you one step back in the running application.
- If you wish to exit the application, navigate to the “main menu” were you are asked to select the data storage type using the “back” command, and then type “exit”

3. Test cases

- Test cases for all the classes can be ran, by running the JUnit test suite named “**AllTests**” in the folder “**test**” or individually by running DataItemTest, RollingStatisticsTest or RollingStatisticsTimestampedTest test cases separately.

4. Bugs and possible improvements

- Whenever an exception is raised while the program is running, you are taken back one or two steps in the program.
- Identified a small bug where in some cases if an exception is caught, the exception is printed after the application output.
- Possible improvement could be to introduce a level of abstraction, as the RollingStatistics and RollingStatisticsTimestamped classes have some common methods and functionality.