Testing 7.

After implementing design patterns and integrating them with the existing code four test classes were created: DataStorageSingletonTest and HealthDataSimulatorTest to test the working of the getInstance() method embedded into singleton pattern. Then, AlertStrategyTest class was created to check the mechanisms of choosing the right algorithm(strategy) dependent on context. AlertDecoratorTest class checks the working of the decorator pattern. Above all that, the suite of old tests were run to check if the code is integrated properly.

Appropriate screenshots:

- ⊘ Test run at 6/9/2024, 5:33:11 PM
  - ⊘ ⬡ testDataTransmissionFailure()
- ⊘ Test run at 6/9/2024, 5:33:03 PM
  - ⊘ ⬡ testGetRecordsNoMatch()
  - ⊘ ⬡ testGetRecordsNoRecords()
  - ⊘ ⬡ testGetRecordsSpecificRecordType
  - ⊘ ⬡ testGetRecordsWithinTimeRange(
- ⊘ Test run at 6/9/2024, 5:32:57 PM
  - ⊘ ⬡ testSingletonInstance()
- ⊘ Test run at 6/9/2024, 5:32:48 PM
  - ⊘ ⬡ testAddMultipleRecords()
  - ⊘ ⬡ testAddPatientData()
  - ⊘ ⬡ testGetAllPatients()
  - ⊘ ⬡ testGetRecordsNoMatch()
  - ⊘ ⬡ testGetRecordsNoRecords()
  - ⊘ ⬡ testGetRecordsWithinTimeRange(
- ⊘ Test run at 6/9/2024, 5:32:40 PM
  - ⊘ ⬡ testSingletonInstance()
- ⊘ Test run at 6/9/2024, 5:32:32 PM
  - ⊘ ⬡ testAlertGeneration()
  - ⊘ ⬡ testRealTimeDataProcessing()

- ✓ Test run at 6/9/2024, 5:32:23 PM >_
  - ✓ ⬡ testOnClose()
  - ✓ ⬡ testOnError()
  - ✓ ⬡ testOnMessage_InvalidMessage()
  - ✓ ⬡ testOnMessage_ValidMessage()
- ✓ Test run at 6/9/2024, 5:32:14 PM
  - ✓ ⬡ testBloodPressureStrategy()
  - ✓ ⬡ testECGAlertStrategy()
  - ✓ ⬡ testBloodOxygenationStrategy()
- ✓ Test run at 6/9/2024, 5:32:07 PM
  - ✓ ⬡ testBloodPressureTrendAlert()
  - ✓ ⬡ testCriticalBloodPressureAlert()
  - ✓ ⬡ testHypotensiveHypoxemiaAlert()
  - ✓ ⬡ testLowBloodSaturationAlert()
  - ✓ ⬡ testRapidDropBloodSaturationAlert()
- ✓ Test run at 6/9/2024, 5:31:59 PM
  - ✓ ⬡ testAlertSimulation()