Akshaya Muralidharan
Aynesh Sundararaj
Bakar Andguladze

Group 6
MS2

Test Cases Report

AllTests.java

This class creates a test suite to run all the other tests one after another in a particular order:

- ConnectionTest
- InteractionTest
- AdditionalTest
- CacheTest

ConnectionTest.java

This class tests error handling and contains 3 tests:

- **testConnectionSuccess()** asserts that connect() method creates connection to localhost and doesn't throw exception.
- **testUnknownHost()** asserts that in case of unknown host, the connect() method throws **UnKnownHostException**.
- **testIllegalPort()** asserts that in case of illegal port, the connect() method throws **IllegalArgumentException**.

AdditionalTest.java

This class contains 3 tests:

- **testKeyPrestentFileSystem()** creates a file with a key of some tuple value of which is not null and asserts that this file exists.
- **testKeyPresentInCaseOfUpdateInFileSystem():** in this test we create a tuple and update it by assigning a new value to existing key. Then we create a fileReader object with the given key as its argument and read the value from it via BufferedReader. The test asserts that the value will be equal to the updated value
- **testKeyNotPresentFileSystem()** asserts that in case of deleted value, the file with the given key won't be created.

InteractionTest()

This class tests **get(), put()** and **update()** methods:

- **testPut()** asserts that if we have some defined <key, value> pair, the put() method executes without any exceptions and its status is ***PUT_SUCCESS***
- **testPutDisconnected()** asserts that if there is no active connection, the put() method will throw an exception.
- **testUpdate()** asserts that if we re-assign a new value to a certain key with put() method, it won't throw any exceptions, its status will be PUT_UPDATE and the value of that key will be the one assigned later.
- **testDelete()** asserts that deleting a tuple doesn't throw exceptions and its status becomes ***DELETE_SUCCESS***
- **testGet()** tests the get() method and asserts that it retrieves the respective value for a key.
- **testGetUnsetValue()** asserts that calling get(key) will update a response status as ***GET_ERROR*** if the value of ***key*** is null.

Akshaya Muralidharan
Aynesh Sundararaj
Bakar Andguladze

CacheTest.java

This file tests the implementation of all three required cache displacement strategies.

We have **FIFO**, **LRU** and **LFU** strategies.

The test cases create the cache objects of respective classes with fixed size (namely 3) and it tests them according to the structure of each strategy.

- **FIFO** strategy should remove the tuple that has been added first.
  testFIFOCache() creates an object of size 3, adds 4 tuples and checks whether the first tuple got removed.
- **LFU** strategy should remove the tuple that has been used least frequently.
  testLFUCache() creates an object of size 3, adds 4 tuples, checks 3 of them and then checks whether the one that hasn't been checked, has been removed.
- **LRU** strategy should remove the tuple that has been used earlier than any other.
  testLRUCache() creates an object of size 3, adds 4 tuples, checks 2 of them and then checks whether the first one has been removed (As it is the one that hasn't been used since creation.)