Test Report

AdditionalTest

In AdditionalTest suite *marshal()* and *unmarshal()* functions of Marshaller class were tested. It was verified, that KVMessage has the same information after being marshaled and unmarshaled.

- ✓ The first test case *testMarshalAndUnmarshalGetMessage()* tests functionality of class Marshaller to proceed KVMessages with StatusType GET. It marshals and unmarshals the message and checks if KVMessage has the same "statusType" and "key" attributes after applying functions *marshal()* and *unmarshal().*
- ✓ The second test case is *testMarshalAndUnmarshalGetSuccessMessage().* It also tests the marshaling and unmarshalling functions of the Marshaller class but for KVMessage with a StatusType GET_SUCCESS. KVMessage expects to have the same "statusType" and "value" attributes as were set before marshaling and unmarshaling.
- ✓ The third test case *testMarshalAndUnmarshalPutMessage()* tests the same functionality as previous two test cases but for StatusType PUT. KVMessage is expected to have unchanged attributes "statusType", "key" and "value".
- ✓ The fourth test case is *testMarshalAndUnmarshalPutSuccessMessage()* must test Marshaller function for KVMessage with StatusType PUT_SUCCESS. Only "statusType" field of KVMessage class must be unchanged, two others are null.

InteractionTest

- ✓ Test case *testPut()* sends key and value to the server and expects KVMessage as a response with StatusType PUT_SUCCESS. No exception must be thrown.
- ✓ In *testPutDisconnected()* connection to the server is closed and the attempt to send message with key and value should cause an Exception.
- ✓ By *testUpdate()* one key value pair is first send to the server. After that a new message with the same key, but different value is put. Response should contain StatusType PUT_UPDATE and latest put value. No exception must be thrown.
- ✓ In *testDelete()* put operation first sends key and value to the server. Another put operation with the same key, but value "null" should delete previous pair of key and value. Received KVMessage has StatusType DELETE_SUCCESS. No exception must be thrown.
- ✓ Test case *TestGet()* should first send put message to server with key and value. The client sends get message with the same key and expects to get value that was set in previous put operation. No exception must be thrown.
- ✓ In test case *testGetUnsetValue()* message get with key is send to the server. Sent key was not previously put to the server, so received message must have StatusType GET_ERROR.
- ✓ We added test for deleting value, that was not previously set. In *testDeleteNonExistingKvPair()* put message with key and value "null" is sent to the server. This key value pair was not put before, that is why server should sent KVMessage with StatusType DELETE_ERROR.

ConnectionTest

- ✓ In *testConnectionSuccess()* connection to the server with given IP and port should be established. No exception must be thrown.
- ✓ In *testUnknownHost()* connection attempt to the unknown server should cause an UnknownHostException.
- ✓ In test case *testIllegalPort()* connection attempt to the invalid port should cause an IllegalArgumentException.