

EditorTest.java

```
* Editor Test Class
import static org.junit.Assert.*;

public class EditorTest {

    @Test
    public void testInsertAfter() {
        Editor e = new Editor();
        e.processCommand("i Test 1");
        e.processCommand("i Test 2");
        assertEquals("1: Test 1\n--> 2: Test 2\n",
            e.getProcess().getList().toString());
    }

    @Test
    public void testInsertBefore() {
        Editor e = new Editor();
        e.processCommand("b Test 1");
        e.processCommand("b Test 2");
        assertEquals("--> 1: Test 2\n 2: Test 1\n",
            e.getProcess().getList().toString());
    }

    @Test
    public void testInsertLast() {
        Editor e = new Editor();
        e.processCommand("b Test 1");
        e.processCommand("b Test 2");
        e.processCommand("e Test 3");
        assertEquals("1: Test 2\n 2: Test 1\n--> 3: Test 3\n",
            e.getProcess().getList().toString());
    }

    @Test
    public void testMoveUp() {
        Editor e = new Editor();
        e.processCommand("b Test 1");
        e.processCommand("b Test 2");
        e.processCommand("e Test 3");
        e.processCommand("u");
        assertEquals("Test 1\n", e.getCurrentLine());
    }

    @Test
    public void testMoveUpMult() {
        Editor e = new Editor();
        e.processCommand("b Test 1");
        e.processCommand("b Test 2");
    }
```

EditorTest.java

```
e.processCommand("e Test 3");
e.processCommand("u 2");
assertEquals("Test 2\n", e.getCurrentLine());
}

@Test
public void testMoveDown() {
    Editor e = new Editor();
    e.processCommand("i Test 1");
    e.processCommand("b Test 2");
    e.processCommand("b Test 3");
    e.processCommand("b Test 4");
    e.processCommand("m");
    assertEquals("Test 3\n", e.getCurrentLine());
}

@Test
public void testMoveDownMult() {
    Editor e = new Editor();
    e.processCommand("i Test 1");
    e.processCommand("b Test 2");
    e.processCommand("b Test 3");
    e.processCommand("b Test 4");
    e.processCommand("m 2");
    assertEquals("Test 2\n", e.getCurrentLine());
}

@Test
public void testRemove() {
    Editor e = new Editor();
    e.processCommand("i Test 1");
    e.processCommand("i Test 2");
    e.processCommand("i Test 3");
    e.processCommand("r");
    assertEquals("1: Test 1\n-> 2: Test 2\n",
        e.getProcess().getList().toString());
}

@Test
public void testRemoveMult() {
    Editor e = new Editor();
    e.processCommand("i Test 1");
    e.processCommand("i Test 2");
    e.processCommand("i Test 3");
    e.processCommand("r 2");
    assertEquals("-> 1: Test 1\n",
        e.getProcess().getList().toString());
}
```

EditorTest.java

```
@Test
public void testDisplay() {
    Editor e = new Editor();
    e.processCommand("i Test 1");
    e.processCommand("i Test 2");
    e.processCommand("i Test 3");
    e.processCommand("d");
    assertEquals("1: Test 1\n2: Test 2\n--> 3: Test 3\n",
        e.getProcess().getList().toString());
}

@Test
public void testDisplaySpecific() {
    Editor e = new Editor();
    e.processCommand("i Test 1");
    e.processCommand("i Test 2");
    e.processCommand("i Test 3");
    String t1 = e.getProcess().getList().display(2,3);
    assertEquals("2: Test 2\n--> 3: Test 3\n", t1);
}

@Test
public void testClear() {
    Editor e = new Editor();
    e.processCommand("i Test 1");
    e.processCommand("i Test 2");
    e.processCommand("i Test 3");
    e.processCommand("!c");
    assertEquals("", e.getProcess().getList().toString());
}

@Test
public void testSaveLoad() {
    Editor e = new Editor();
    e.processCommand("i Test 1");
    e.processCommand("i Test 2");
    e.processCommand("i Test 3");
    e.processCommand("s file1");
    Editor f = new Editor();
    f.processCommand("!l file1");
    f.processCommand("m 2");
    assertEquals(e.getProcess().getList().toString(),
        f.getProcess().getList().toString());
}

@Test
public void testExit() {
```

EditorTest.java

```
Editor e = new Editor();
e.processCommand("!x");
assertFalse(e.getActive());
}

@Test
public void testCut() {
    Editor e = new Editor();
    e.processCommand("i Test 1");
    e.processCommand("cut 1 1 1");
    assertEquals("", e.getProcess().getList().toString());
}

@Test
public void testPaste() {
    Editor e = new Editor();
    e.processCommand("i Test 1");
    String t1 = e.getProcess().getList().toString();
    e.processCommand("cut 1 1 1");
    e.processCommand("pas 1");
    assertEquals(t1, e.getProcess().getList().toString());
}

@Test
public void testHelp() {
    Editor e = new Editor();
    String t1 = e.getProcess().showHelp();
    String t2 = "";
    t2 += ("Welcome to Text Editor Help!\n");
    t2 += ("Command:      Function:\n");
    t2 += ("b 'sentence'  Insert sentence before current");
    t2 += (" line, make inserted line current\n");
    t2 += ("i 'sentence'  Insert sentence after current");
    t2 += (" line, make inserted line current\n");
    t2 += ("e 'sentence'  Insert sentence after last line, make inserted
line current\n");
    t2 += ("m              Move cursor down 1 position\n");
    t2 += ("m #           Move cursor down # positions\n");
    t2 += ("u              Move cursor up 1 position\n");
    t2 += ("u #           Move cursor down # positions\n");
    t2 += ("r              Remove current line. Next line becomes ");
    t2 += ("current, unless no next \n          line, then previous
becomes ");
    t2 += ("current\n");
    t2 += ("r #           Remove # lines, starting at current\n");
    t2 += ("d              Display all lines with line numbers\n");
    t2 += ("d # *         Display lines # to * with line numbers\n");
    t2 += ("cut # $ *     Cut lines # to $ to clipboard *\n");
}
```

EditorTest.java

```
t2 += ("pas *      Paste clipboard * before current position\n");
t2 += ("c          Clear all lines in the file\n");
t2 += ("!c         Force clear all lines in the file\n");
t2 += ("s 'filename' Save contents to specified text file\n");
t2 += ("l 'filename' Load contents of file into current buffer\n");
t2 += ("!l 'filename' Force load contents of file into current buffer\n");
\n");
t2 += ("h          Display this help page\n");
t2 += ("x          Exit the editor\n");
t2 += ("!x         Force exit the editor");
assertEquals(t1,t2);
}
```