|  |  |
| --- | --- |
| **Project Name: Project 1: Voting System Team# 19** | |
| **Test Stage: Unit \_x\_ System \_\_** | **Test Date: 3/13/2021** |
| **Test Case ID#: 2** | **Name(s) of Testers: Isaiah Herr** |
| **Test Description:**  **Test fileParser class functions** |  |
| **Automated: yes\_\_x\_ no \_\_\_** | **Indicate where are you storing the tests (what file) and the name of the method/functions being used.**  UnitTester.java |
| **Results: Pass \_\_x\_\_\_ Fail\_\_\_\_\_\_\_\_** |  |
|  |  |
| **Preconditions for Test:**  Must have strings indicating IRBallots or OPLBallots. Must also have a string to represent a candidate.  Must create an array of candidates. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step**  **#** | **Test Step**  **Description** | **Test**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Notes** |
|  |  |  |  |  |  |
| 1 | Test createParties() and see if it successfully parses names and parties into the correct variables for the Party class and candidate class | Candidates:  new Candidate(“Bob”, “C”, 0, 0, false),  new Candidate("Tom", "R", 0, 0, false),  new Candidate("Isabelle", "R", 0, 0, false),  new Candidate("Bill", "C", 0, 0, false),  new Candidate("Maddie", "D", 0, 0, false),  new Candidate("Ralph", "R", 0, 0, false),  new Candidate("Rolando", "D", 0, 0, false) | Should return with no errors  Should also print out The party name along with any candidates associated with that party. Example is below:  Party: Republican   * Tom * Billy   Party: Democrat   * Joey * Sandra | No errors. Also prints out:  Party: C   * Bob, C * Bill, C   Party: R   * Tom, R * Isabelle, R * Ralph, R   Party: D   * Maddie, D * Rolando, D | Created candidates successfully assuming that file has been parsed properly |
| 2 | Test Fileparser.getBallot() for OPL elections using strings indicating ballots  Strings should look similar to this and should not differ from this format:  “,,,1,,”  “1,,,,,”  “,,1,,,” | String ballotLine1 = ",,,1,,";  String ballotLine2 = "1,,,,,";  String ballotLine3 = ",,,,,1";  OPLBallot ballot1 = (OPLBallot) FileParser.*getBallot*(ballotLine1, 1, 1);  OPLBallot ballot2 = (OPLBallot) FileParser.*getBallot*(ballotLine2, 1, 2);  OPLBallot ballot3 = (OPLBallot) FileParser.*getBallot*(ballotLine3, 1, 3); | Should successfully create an OPLBallot object with its own ballotNum and have the correct index for the vote from the ballot that is parsed from.  Should receive the index from each ballot as:  3  0  5  and not return any errors | OPLBallot ballot1(3, 1)  OPLBallot ballot2(0, 2)  OPLBallot ballot3(5, 3)  Also retrieves vote index from each ballot successfully as well and returns  3  0  5  For each respective ballot with no errors | Test completed successfully |
| 3 | Test.getBallot() for IR elections using strings indicating the ballot  Strings should look similar to this and should not differ in this format:  “1,2,3,4,5”  “2,3,1,5,4”  “2,3,1,,4” | String iBallotLine1 = "1,5,3,2,4";  String iBallotLine2 = "4,3,2,1,5";  String iBallotLine3 = ",2,,1,";  IRBallot iBallot1 = (IRBallot) FileParser.*getBallot*(iBallotLine1, 0, 1);  IRBallot iBallot2 = (IRBallot) FileParser.*getBallot*(iBallotLine2, 0, 2);  IRBallot iBallot3 = (IRBallot) FileParser.*getBallot*(iBallotLine3, 0, 3);  int[] ranks1 = iBallot1.getRanks();  int[] ranks2 = iBallot2.getRanks();  int[] ranks3 = iBallot3.getRanks(); | Should successfully create an IRBallot and then put the integer arrays from each IRBallot into a new integer array variable for testing. Should receive the integer array indicating the preferences of a candidate when using getRanks() for each ballot as shown:  [1, 5, 3, 2, 4]  [4 ,3 ,2 , 1, 5]  [-1, 2, -1, 1, -1] | Successfully creates IRBallot objects:  IRBallot([1, 5, 3, 2, 4], 1)  IRBallot([4, 3, 2, 1, 5], 2)  IRBallot([-1, 2, -1, 1, -1], 3)  When using getBallot for each of these ballots, the integer array lines up with the information in the string and successfully returns true for  *assertArrayEquals*(new int[]{1,5,3,2,4}, ranks1);  *assertArrayEquals*(new int[]{4,3,2,1,5}, ranks2);  *assertArrayEquals*(new int[]{-1,2,-1,1,-1}, ranks3); | Test completed successfully |
| 4 | Test candidate creation for IRElections using FileParser.getCandidates()  Should be able to assign the correct names and party to a candidate object when parsed from a string | String IRCandidateLine = "Rosen (D), Kleinberg (R), Chou (I), Royce(L)";  Candidate[] candidatesIR = FileParser.*getCandidates*(IRCandidateLine,0);  *assertEquals*(candidatesIR[0].getName(), "Rosen");  *assertEquals*(candidatesIR[0].getParty(), "D"); | Should be able to successfully create a Candidate class and correctly associate the name and party to that specific candidate inserted into a candidate array  Expected results:  First IR Candidate should be Rosen  “Rosen”  “D”  These should return with no errors when using the getters from candidate class | “Rosen”  “D”  returns and asserts successfully with no errors for  *assertEquals*(candidatesIR[0].getName(), "Rosen");  *assertEquals*(candidatesIR[0].getParty(), "D"); | Tests passed successfully |
| 5 | Test candidate creation for OPLElections using FileParser.getCandidates()  Should be able to assign the correct names and party to a candidate object when parsed from a string | String OPLCandidateLine = "[Pike,D], [Foster,D],[Deutsch,R], [Borg,R], [Jones,R],[Smith,I] ";  Candidate[] candidatesOPL = FileParser.*getCandidates*(OPLCandidateLine,1);  *assertEquals*(candidatesOPL[2].getName(), "Deutsch");  *assertEquals*(candidatesOPL[2].getParty(), "R"); | Should successfully create candidates for an OPL election and inserted into a candidate array.  Should return true for all asserts when getting name and party from a specific candidate in the array  Expected results:  “Deutsch”  “R” | “Deutsch”  “R”  returns and asserts successfully for  *assertEquals*(candidatesOPL[2].getName(), "Deutsch");  *assertEquals*(candidatesOPL[2].getParty(), "R"); | Completed tests |

**Post condition(s) for Test:**



Should return true for all asserts and successfully parsed strings to create candidates and ballots for their respective elections