Voting System
Spring Backlog

Team 1

- Allison Miller (mill7079)
- Jake Waro (warox001)
- Sami Frank (fran0942)
- Declan Buhrsmith (buhrs001)

Test 1: Checking invalidated ballots are written to invalidated ballots report
Team Member(s) Responsible: Jake
Inputs: **1.** A .csv file containing an election's candidates on the first line, and then each subsequent line represents a ballot.
Tests: **1.** Test with files with one invalid ballot. **2.** Test with files with many invalid ballots. **3.** Test with files with odd number candidates. **4.** Test with files with even number candidates. **5.** Test with files with odd number ballots. **6.** Test with files with even number ballots. **7.** Test with files with no invalid ballots.
Outputs: Invalidated_Ballots.txt contains a list of invalidated ballots, which file they were found in, and the line number they are at in the input file.
Passed or Failed: Passed
Date: April 21, 2020

Test 2: Checking isValid method in Election class (even number of candidates)
Team Member(s) Responsible: Sami
Inputs: **1.** A list of strings representing a STV ballot with an even number of candidates. inputList = ["1,2,3,4", "1,,,", ",,1,2", ",1,2,3", ",2,1,", ",,,1"]
Tests: **1.** AssertsTrue on the isValid method for each string in inputList (a total of 6 times)
Outputs: **1.** Pass

2. Fail
3. Pass
4. Pass
5. Pass
6. Fail

Where each numbered output refers to each respective string in inputList. Numbers 2 and 6 should have failed since they are invalid ballots.

Passed or Failed: Passed

Date: April 21, 2020

Test 3: Checking isValid method in Election class (odd number of candidates)

Team Member(s) Responsible: Sami

Inputs:

1. A list of strings representing a STV ballot with an odd number of candidates. inputList = ["1,2,,,", "1,2,3,,", "1,2,3,4,5", ",,,,1"]

Tests:

1. AssertsTrue on the isValid method for each string in inputList (a total of 4 times)

Outputs:

1. Fail
2. Pass
3. Pass
4. Fail

Where each numbered output refers to each respective string in inputList. Numbers 1 and 4 should have failed since they are invalid ballots.

Passed or Failed: Passed

Date: April 21, 2020

Test 4: Checking conditional in createBallotList method in Election class (if election is type plurality)

Team Member(s) Responsible: Sami

Inputs:

1. A list of files representing plurality ballots.

Tests:

1. Creates new election with input files.
2. Creates a ballot list based on input files.
3. Hits the conditional stating the ballot is type plurality and will create a ballot rather than checking to see if its valid before doing so.
4. Gets LinkedHashSet<Ballot> after the ballot list has been formed.
5. Confirms that the number of ballots matches the length of the ballot list

Outputs:
All ballots would be added to createBallotList since we can assume fair play and all plurality ballots are valid.

Passed or Failed: Passed

Date: April 21, 2020

Test 5: Checking conditional in createBallotList method in Election class (if election is type STV)

Team Member(s) Responsible: Sami

Inputs:
1. A list of files representing STV ballots.

Tests:
1. Creates new election with STV input files.
2. Creates a ballot list based on input files.
3. Within the createBallotList, it is determined to check if the ballot is valid.
4. Gets LinkedHashSet<Ballot> after the ballot list has been formed.
5. Confirms that the number of ballots matches the length of the ballot list

Outputs:
A ballot list has been created and there has been checks to determine whether or not each ballot was valid.

Passed or Failed: Passed

Date: April 21, 2020

Test 6: Checking if GUI prompt works with existing UI.

Team Member(s) Responsible: Declan

Inputs: **2.** The prompt used in the CLI UI to pull up the GUI. This would be the string “f”
Tests: AssertEquals on the quitProgram boolean that returns false if the program would correctly pull-up the user interface, and returns false if for some reason the prompt, that being f is not able to be reached.
Outputs: No output, a user would then select files from the GUI so on a real run the output would be a new Java Applet window that contains the file-picker user interface.
Passed or Failed: Passed
Date: April 27, 2020

Test 7: Checking that runFileGUI returns the files selected in the GUI
Team Member(s) Responsible: Allison
Inputs: 1. User interaction selecting files
Tests: 1. User selects files, checks output file to see if the file names are correct 2. User presses cancel or closes the window and the test ensures null is returned
Outputs: test_getFiles in testing/testOutputs file lists all files selected by the user.
Passed or Failed: Passed
Date: April 26, 2020

Test 8: Checking that all files are displayed in the GUI
Team Member(s) Responsible: Allison
Inputs: 1. User interaction
Tests: 1. User opens a folder in the file chooser window and visually confirms that all files are displayed that are present and visible in the directory.
Outputs: No output

Passed or Failed: Passed
Date: April 26, 2020