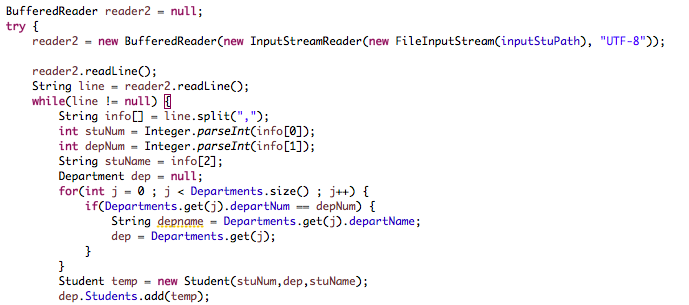
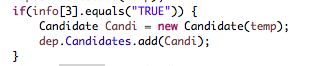
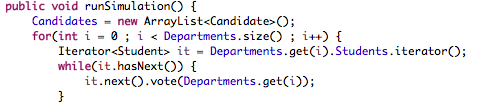
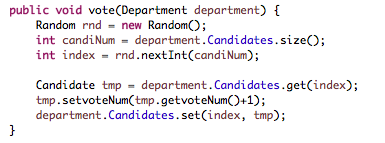
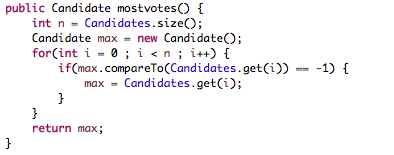
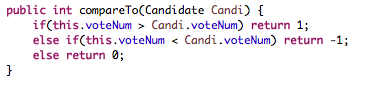
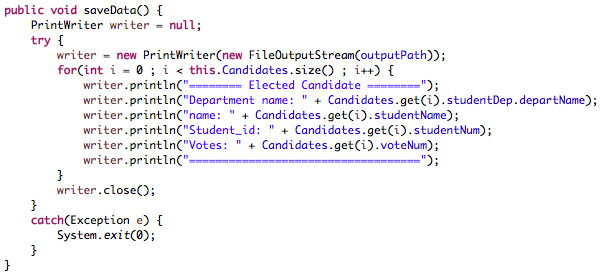
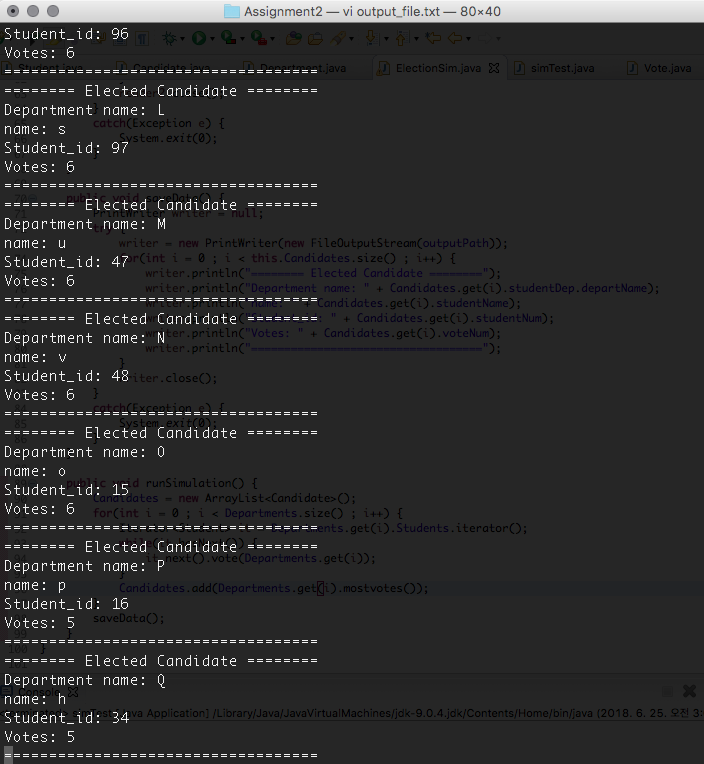


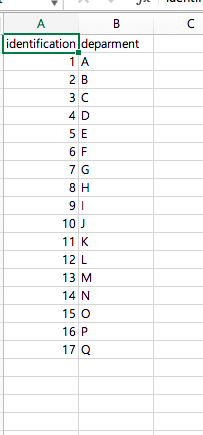
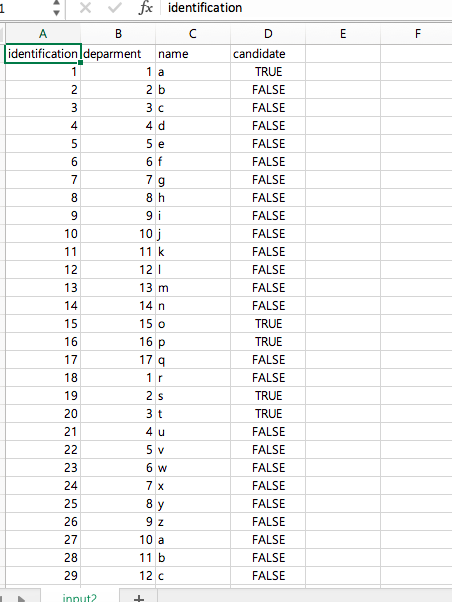


1. ElectionSim 클래스의 생성자에서 input1.csv파일을 열어 ArrayList인 Departments에 각 학과의 이름과 번호에 대한 정보를 가지는 Department객체를 저장한다.



1. Input2.csv파일을 열어 각 학생들이 어느 학과인지 Departments를 순회하며 알아내고, 알아낸 학과 이름과 학생이름, 학생번호를 그 Department 객체의 ArrayList인 Students에 저장한다.  
     
   
2. 이때, 후보자인 학생이면 그 Department 객체의 ArrayList인 Candidates에도 저장한다.  
     
   
3. Departments의 수만큼 for문을 돌리는데, 각 Department 객체의 ArrayList인 Students를 순회하며, 각 학생마다 vote(Department dep)을 실행한다.  
   
4. vote(Department dep) 안에서 각 학생들은 각 과의 후보자들 중 한 명에게 랜덤으로 투표를 하게 되는데, 각 Department 객체의 Candidates 사이즈로 난수를 생성하여 해당 후보자 학생의 득표수를 getter로 가져와 1을 더해 setter로 득표수를 수정해준다. 그리고 이 후보자의 객체를 다시 Candidates에 set메소드를 이용하여 대체한다.  
     
   
5. 각 학과의 Students를 모두 순회하면, 모든 학생들의 투표가 끝난 것이므로 해당 학과의 Candidates에서 제일 득표수가 많은 후보를 골라낸다. 이는, mostvotes()로 구현하였다. Comparable 인터페이스에서 재정의한 compareTo메소드를 이용하여 제일 득표수가 많은 후보자를 반환한다.  
     
   
6. compareTo메소드에서 만약 비교대상보다 득표수가 많으면 1, 적으면 -1, 같으면 0을 반환하도록 하여 max값을 계속해서 갱신하고 마지막에 제일 득표수가 많은 후보자인 max를 반환한다.  
     
   
7. 각 학과마다 득표수가 제일 많아 선출된 후보자들은 Candidates라는 ArrayList에 넣는다.  
     
   
8. saveData() 메소드에서 “output\_file.txt”라는 파일을 생성하여 Candidates를 순회하며 각 학과마다 선출된 후보자의 정보를 작성한다.



<input1.csv> <input2.csv>