



Object – Oriented System Design

Programming Assignment #03

● Programming Assignment

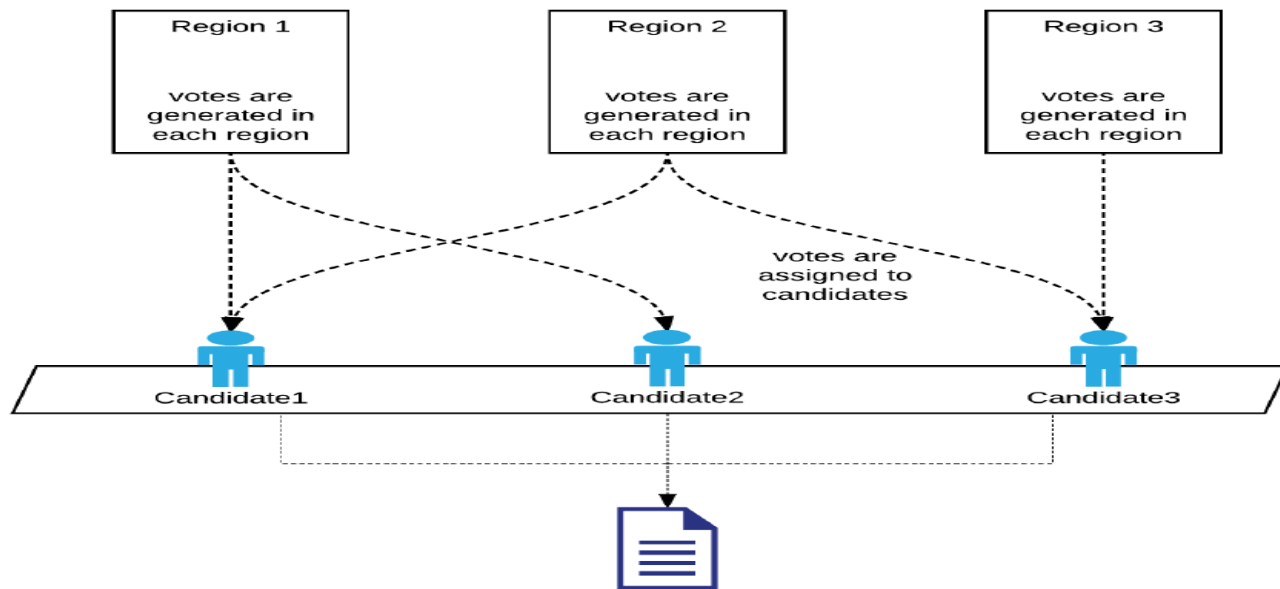
- Template : Assignment03.
- 업로드 : 각자의 프로젝트 폴더 (기존의 Self-Test업로드 방식과 동일)
- Commit Message : Assignment03 (git commit -m "Assignment03")
- 제출기한 : 6월 5일 밤 12시까지

CSLAB

● 투표 시뮬레이터

● 투표 시뮬레이터

- 본 과제에서는 선거 경쟁의 결과를 예측하는 투표 시뮬레이터를 개발한다. 실제 투표가 아닌 시뮬레이션 이므로 유권자가 어떤 후보를 투표할지는 랜덤으로 결정한다.
- 본 과제에서는 Information Hiding 개념을 반드시 지켜야 하나, 필요한 경우 Privacy Leak 문제는 무시해도 무관하다. 또한, 필요한 경우 Template에 variable 및 method 추가/수정이 가능하다.
- 투표 시뮬레이터의 구조는 아래 그림과 같다.



● Candidate 클래스

- 각 후보자의 정보를 저장하는 Candidate 클래스를 만든다. Candidate 클래스는 3개의 instance variables를 가진다
 - String name : 후보자의 이름
 - int numVotes : 후보자가 받은 투표 수
 - Votes[] votes : 후보자가 받은 표의 배열
- 2개의 인자 (String name, int maxVotes)를 받는 생성자를 만든다
 - 첫번째 인자 name을 후보자의 이름으로 설정한다
 - Vote 타입의 배열을 생성한 뒤 크기를 두번째 인자인 maxVotes로 설정한다. 후보자가 받을 수 있는 최대 득표 수를 나타낸다.
- Candidate 클래스는 Comparable을 구현(implements)한다. 후보자들 간의 비교(Compare)는 후보자가 받은 투표 수인 numVotes에 기반하며, 정렬은 오름차순으로 이루어진다.

CSLAB

● Candidate 클래스

- toString메서드는 다음과 같은 형태의 문자열을 반환한다.

```
-----Candidate-----  
Name: Ashley  
Total Votes: 589
```

- toRegionString() 메서드는 int regionNum을 입력으로 받아 해당 region에서의 득표수를 반환한다.
- Candidate 클래스의 inner class 인 Vote클래스를 생성한다. Vote클래스는 private로 선언되어야 한다.
 - Vote 클래스는 후보자가 투표를 받을때마다 votes배열을 늘려준다.
 - Vote 클래스는 하나의 instance variable (int regionNum : 지역번호)을 가진다.
 - 1개의 인자 (int regionNum)을 인자로 받는 생성자를 만든다.
 - int regionNum을 인자로 받는 addVote()메서드를 만든다. 해당 메서드는 Vote객체를 생성한 뒤 해당 객체를 후보자의 votes배열에 추가한다.
 - addVote()메서드는 여러 쓰레드가 동시에 특정 자원에 접근하므로 반드시 동기화가 이루어져야 함.
- Vote 클래스의 addVote() 메서드를 호출하는 callAddVotes() 메서드를 Candidate 클래스에 추가한다. 해당 메서드는 int regionNum을 인자로 받는다. (투표권을 가지고 있는게 아니라 투표받은 개수)

● Region 클래스

- 투표 지역구에 대한 정보를 저장하는 Region 클래스를 생성한다. Region 클래스는 쓰레드를 사용하므로 Runnable interface를 구현(implements)하거나 Thread class를 상속(extends)해야한다.
- Region 클래스는 4개의 instance variables를 갖는다
 - String regionName : 지역구 이름
 - int regionNum : 지역번호
 - int population : 지역 인구
 - Candidate[] candidates : 선거 후보자 배열
- 모든 instance variable을 인자로 받는 생성자를 만든다
- generateVotes() 메서드를 만든다. 메서드는 다음과 같이 구현해야 한다.
 - 해당 메서드를 통해 특정 지역구에 있는 유권자가 랜덤한 후보자를 투표한다.
 - 랜덤한 후보자를 투표하기 위해 Math 혹은 Random 클래스를 활용하여 0 ~ 후보자 수 범위의 랜덤한 숫자를 생성한다.
 - 선택한 랜덤한 후보자 객체를 통해 callAddVotes()메서드를 호출하여 해당 후보자를 투표한다.

CSLAB

● ElectionSim 클래스

- 선거 시뮬레이터로 사용될 ElectionSim 클래스를 생성한다. 해당 클래스는 입력 텍스트 파일을 읽어 들여 저장한 뒤 선거 시뮬레이션을 실행한다.
- ElectionSim 클래스는 4개의 instance variable을 가진다.
 - String outputFile : 출력 파일의 경로
 - int population : 전 지역의 총합 인구 수 (최대 투표 수)
 - Candidate[] candidates : 후보자 명단 배열
 - Region[] regions : 선거 지역구 배열
- 두 개의 인자 (String inputFile, String outputFile)를 받는 생성자를 생성한다.
 - String inputFile : 입력 파일 경로
 - String outputFile : 출력 파일 경로
 - 생성자는 다음과 같은 동작을 수행해야 한다.
 - instance variable 중 String outputFile을 설정한다.
 - 입력 받은 inputFile을 통해 다음과 같은 동작을 수행
 - instance variable 중 population을 설정한다.
 - Candidate 클래스 객체를 생성하고 instance variable 중 candidates 배열에 추가한다.
 - Region 클래스 객체를 생성하고 instance variable 중 regions배열에 추가한다.
 - 입력 파일의 형태는 오른쪽 그림과 같다.

```
POPULATION 4300  
CANDIDATES 8  
Mickey  
Mia  
Anthony  
Katy  
Lewis  
Ashley  
Danny  
Karen  
REGIONS 5  
Seoul 1 1500  
Daegu 2 700  
Daejeon 3 300  
Gwangju 4 800  
Busan 5 1000
```

● ElectionSim 클래스

- **saveData() 메서드를 생성한다.**
 - candidates 배열을 정렬한다
 - 출력파일에 선거 결과를 입력한다 (출력 형식은 마지막 페이지 그림 참조)
- **startSimulation() 메서드를 생성한다. 해당 메서드는 선거 시뮬레이션을 시작한다.**
 - 모든 생성된 regions들을 시작해야한다.
 - regions에 대해 시작된 모든 쓰레드들이 종료될 때까지 대기해야 한다. (Thread 관련 적절한 메서드를 활용할 것)
 - 모든 쓰레드들이 종료되면 saveData() 메서드를 호출하여 선거 결과를 출력 파일에 기록한다.

CSLAB

● SimTest 클래스

- 메인 메서드를 수행하는 클래스
- 입력 파일 경로와 출력 파일 경로가 상수로 고정되어 있다.
- 입력 파일 경로와 출력 파일 경로를 인자로 받아 ElectionSim 클래스 객체를 생성한 뒤 startSimulation() 메서드를 호출한다.

CSLAB

● 출력 결과 예시

```

outputfile.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

-----Candidate-----
Name: Mickey
Total Votes: 559
Seoul: 201
Daegu: 93
Daejeon: 36
Gwangju: 95
Busan: 134

=====
-----Candidate-----
Name: Ashley
Total Votes: 548
Seoul: 197
Daegu: 91
Daejeon: 43
Gwangju: 93
Busan: 124

=====
-----Candidate-----
Name: Mia
Total Votes: 546
Seoul: 188
Daegu: 90
Daejeon: 39
Gwangju: 103
Busan: 126

=====
-----Candidate-----

```

```
-----Candidate-----
```

```

Name: Katy
Total Votes: 538
Seoul: 190
Daegu: 96
Daejeon: 29
Gwangju: 109
Busan: 114

```

```
=====
-----Candidate-----
```

```

Name: Karen
Total Votes: 537
Seoul: 179
Daegu: 74
Daejeon: 44
Gwangju: 113
Busan: 127

```

```
=====
-----Candidate-----
```

```

Name: Lewis
Total Votes: 535
Seoul: 188
Daegu: 72
Daejeon: 43
Gwangju: 95
Busan: 137

```

```
=====
```

```
=====
-----Candidate-----
```

```
-----Candidate-----
```

```

Name: Anthony
Total Votes: 530
Seoul: 198
Daegu: 82
Daejeon: 26
Gwangju: 101
Busan: 123

```

```
=====
-----Candidate-----
```

```
-----Candidate-----
```

```

Name: Danny
Total Votes: 507
Seoul: 159
Daegu: 102
Daejeon: 40
Gwangju: 91
Busan: 115

```