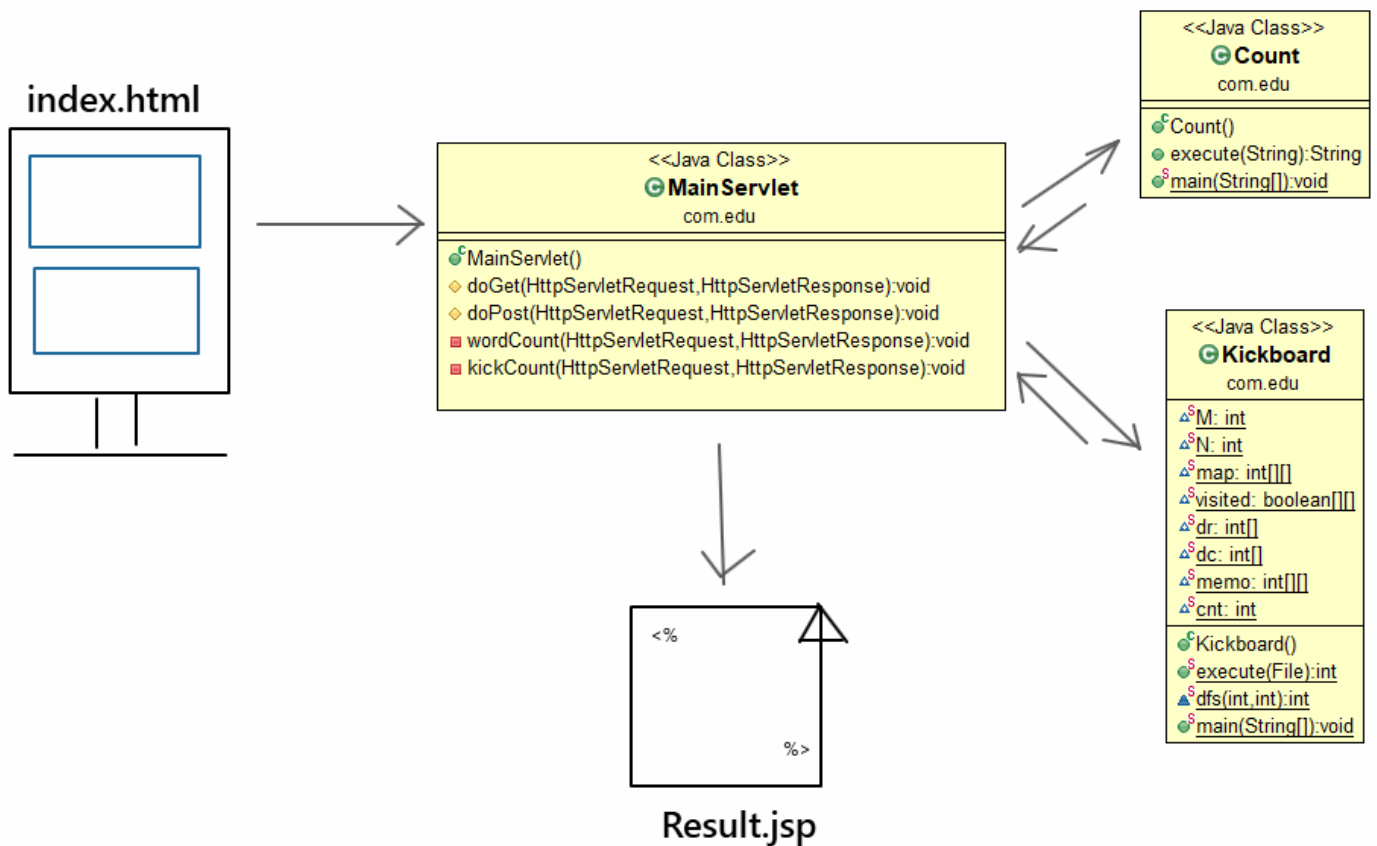
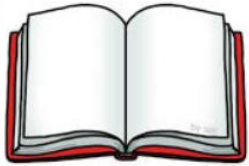


- ✓ 알고리즘을 이용해서 문제를 해결하고 이를 웹 프로그램 결과로 표현할 수 있다.
- ✓ 비동기 통신의 요청, 응답을 핸들링 할 수 있다.
- ✓ 페이지의 동적 처리를 JQuery를 이용해서 작성할 수 있다.

### 프로그램 전체 흐름도





- ✓ 알고리즘을 이용해서 문제를 해결하고 이를 웹 프로그램 결과로 표현할 수 있다.
- ✓ 비동기 통신의 요청, 응답을 핸들링 할 수 있다.
- ✓ 페이지의 동적 처리를 JQuery를 이용해서 작성할 수 있다.

## | 구현 해야 할 내용 1. 단어 세기 (Count.java)

<<Java Class>>	
G Count	
com.edu	
Count()	
execute(String):String	
main(String[]):void	

위 Count 클래스의 execute()를 구현하세요.

execute()는 파라미터로 문장이 입력된다. 그 문장내에 가장 많이 등장하는 단어를 찾아 리턴하세요.

<조건>

- 가장 많이 등장하는 단어의 빈도수가 같을 경우 알파벳순으로 앞선 단어를 출력한다.
- 모두 소문자로 변경하여 처리한다.

파라미터 예)

Can Danny and his father outsmart the villainous Mr. Hazell? Danny has a life any boy would love - his home is a gypsy caravan, he's the youngest master car mechanic around, and his best friend is his dad, who never runs out of wonderful stories to tell. But one night Danny discovers a shocking secret that his father has kept hidden for years.

리턴 예)

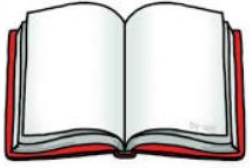
his

파라미터 예)

I like cat. I like cat. I like cat.

리턴 예)

cat.



- ✓ 알고리즘을 이용해서 문제를 해결하고 이를 웹 프로그램 결과로 표현할 수 있다.
- ✓ 비동기 통신의 요청, 응답을 핸들링 할 수 있다.
- ✓ 페이지의 동적 처리를 JQuery를 이용해서 작성할 수 있다.

## | 구현 해야 할 내용 2.세준이의 킥보드 (Kickboard.java)

<<Java Class>>	
Kickboard com.edu	
△ <sup>S</sup> M: int	
△ <sup>S</sup> N: int	
△ <sup>S</sup> map: int[][]	
△ <sup>S</sup> visited: boolean[][]	
△ <sup>S</sup> dr: int[]	
△ <sup>S</sup> dc: int[]	
△ <sup>S</sup> memo: int[][]	
△ <sup>S</sup> cnt: int	
Kickboard()	
execute(File):int	
dfs(int,int):int	
main(String[]):void	

위 Kickboard 클래스의 execute( )를 구현하세요.

execute( )는 파라미터로 데이터 파일명이 입력된다. 그 파일에 있는 데이터를 읽어서 아래의 내용에 맞게 처리하도록 구현합니다. (추가 메소드를 구현해도 된다.)

### ➤ 문제 설명

킥보드가 갖고 싶었던 동은이는 STAR 매장에서 저렴하게 킥보드를 구매하였다. 선릉 멀캠 교육장 인근을 나타내는 지도는 각 칸마다 지대의 높이가 적혀있고 각 지점을 상하좌우로 이동이 가능하다.

지도에서 가장 왼쪽 위 칸의 바나프레소를 출발하여 가장 오른쪽 아래의 역삼역으로 이동하고자 한다.

그런데 STAR 매장에서 구매한 킥보드는 힘이 약해서 현재 위치보다 더 높이가 낮은 위치로만 이동이 가능하다.

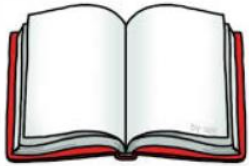
아래 보이는 맵 지도에서는 3가지 이동이 가능하다. 지도가 주어질 때 동은이가 바나프레소에서 역삼역까지 구매한 킥보드를 이용해서 이동할 수 있는 경로의 수를 구해보자.

50	45	37	32	30
35	50	40	20	25
30	30	25	17	28
27	24	22	15	10

50	45	37	32	30
35	50	40	20	25
30	30	25	17	28
27	24	22	15	10

50	45	37	32	30
35	50	40	20	25
30	30	25	17	28
27	24	22	15	10

50	45	37	32	30
35	50	40	20	25
30	30	25	17	28
27	24	22	15	10



- ✓ 알고리즘을 이용해서 문제를 해결하고 이를 웹 프로그램 결과로 표현할 수 있다.
- ✓ 비동기 통신의 요청, 응답을 핸들링 할 수 있다.
- ✓ 페이지의 동적 처리를 JQuery를 이용해서 작성할 수 있다.

#### 입력조건

첫 줄에는 지도의 세로크기  $h$ 와 가로크기  $w$ 가 빈 칸을 사이에 두고 주어진다.  
이어 다음  $h$ 개 줄에 걸쳐 한 줄에  $w$ 개씩 위에서부터 차례로 각 지점의 높이가 빈 칸을 사이에 두고 주어진다.  $h$ 와  $w$ 는 각각 500 이하의 자연수이고 각 지점의 높이는 10000 이하의 자연수이다.

#### 출력조건

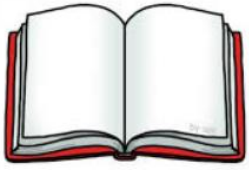
첫 째 줄에 이동 가능한 경로의 수  $k$ 를 출력한다.  
모든 입력에 대하여  $k$ 는 10억 이하의 음이 아니 자연수이다.

입력 예

```
4 5
50 45 37 32 30
35 50 40 20 25
30 30 25 17 28
27 24 22 15 10
```

출력 예

```
3
```



- ✓ 알고리즘을 이용해서 문제를 해결하고 이를 웹 프로그램 결과로 표현할 수 있다.
- ✓ 비동기 통신의 요청, 응답을 핸들링 할 수 있다.
- ✓ 페이지의 동적 처리를 JQuery를 이용해서 작성할 수 있다.

### | 구현 해야 할 내용 3. 웹페이지 구현하기 ( index.html )

#### 단어세기

입력 문자열 :

실행

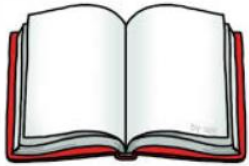
가장 많이 나온 단어 :

#### Kickboard

입력 파일명:

실행

갈수 있는 경로의 수 :



- ✓ 알고리즘을 이용해서 문제를 해결하고 이를 웹 프로그램 결과로 표현할 수 있다.
- ✓ 비동기 통신의 요청, 응답을 핸들링 할 수 있다.
- ✓ 페이지의 동적 처리를 JQuery를 이용해서 작성할 수 있다.

### | 구현 해야 할 내용 3. 웹페이지 구현하기 ( index.html )

#### 실행 결과 화면

### 단어세기

입력 문자열 :

가장 많이 나온 단어 : **cat.**

### Kickboard

입력 파일명:

갈수 있는 경로의 수 : **3**

위처럼 실행되기 위해서  
Ajax, JQuery 기술을 활용한다.