



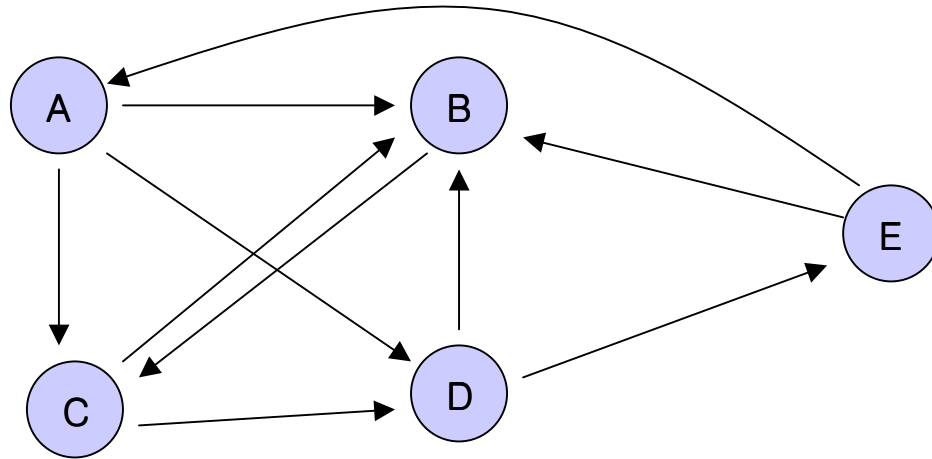
Google™

# PageRank 구현 설명

2007.01

마이엔진 사장 이현봉

# 페이지랭크를 계산할 웹 모양과 main() 코드



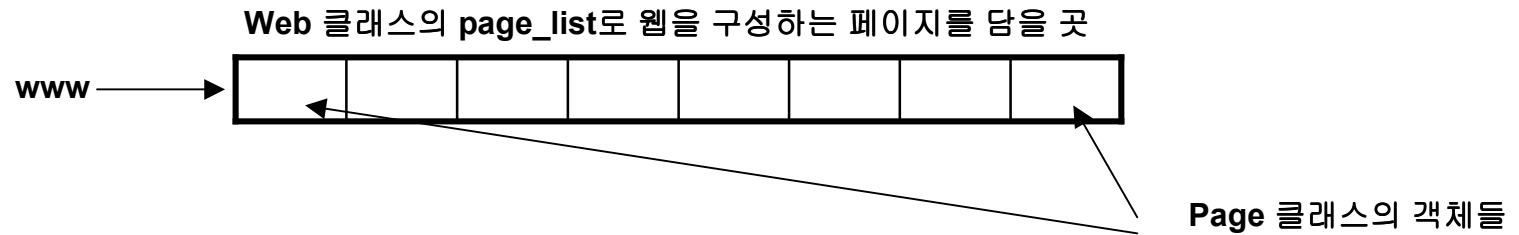
```
public static void main(String[] args)
{
    Web www = new Web();
    Page A = new Page("A", 100.0);
    Page B = new Page("B", -991.0, A);
    Page C = new Page("C", 1.0, A, B);
    Page D = new Page("D", 1.0, C, A);
    Page E = new Page("E", 1.0, D);
    A.insert_bl(E);
    B.insert_bl(C, D, E);

    www.insert_pages_build_web(A, B, C, D, E); // Now web has pages
    www.display_webinfo(); // 확인차 출력해 봄
    System.out.println("");

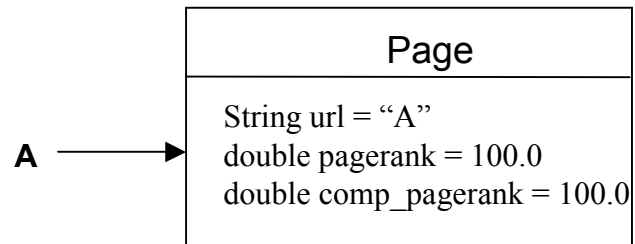
    int iterations = 200; // 몇 번 돌릴까?
    for (int i=0; i<iterations; i++)
    {
        www.calc_pagerank();
        System.out.print(" " + (i+1) + " : ");
        www.display_PageRanks();
        //www.update_pr_values();

        if (i < iterations - 1)
            www.update_pr_values();
    }
    System.out.println("");
    www.display_webinfo();
}
}
```

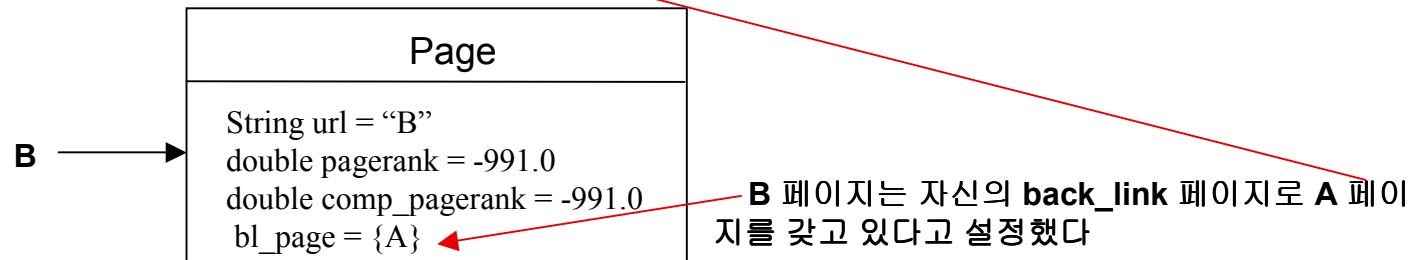
- 1) 일단 웹 객체를 생성한다. `Web www = new Web();`



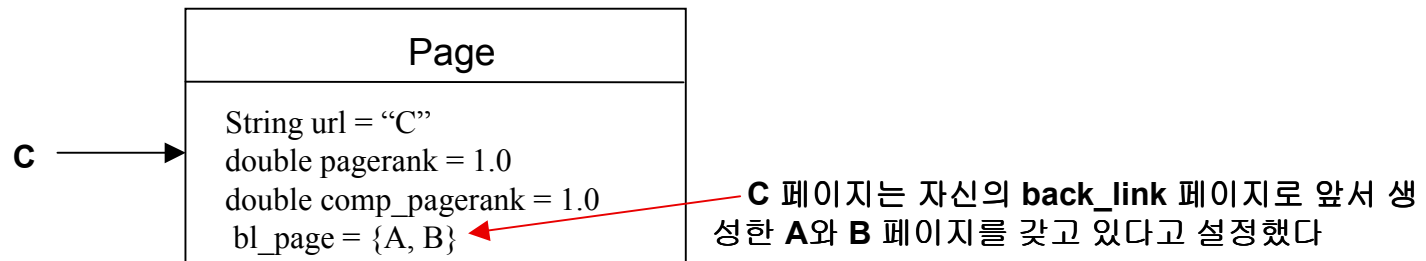
- 2) A 페이지를 만든다. `Page A = new Page("A", 100.0)`. 페이지 인스턴스 A를 생성해 그 url명을 A라 하고, 페이지랭크초기값을 100.0으로 설정한다.



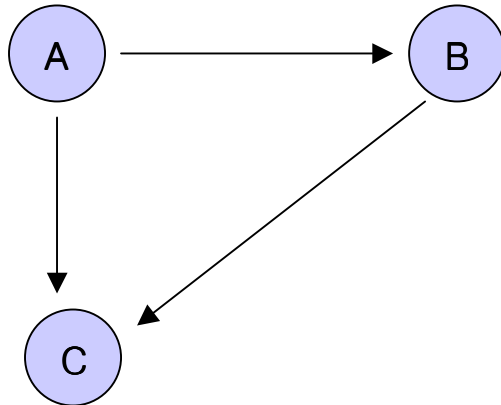
- 3) B 페이지를 만든다. `Page B = new Page("B", -991.0, A)`



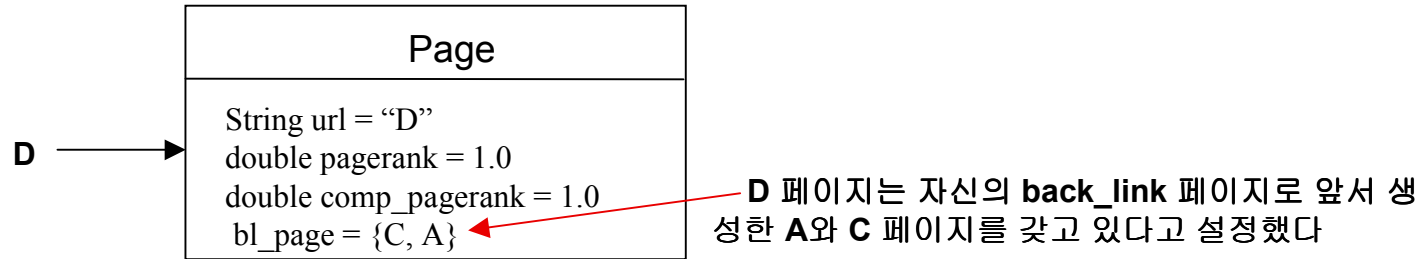
4) C 페이지를 만든다. `Page C = new Page("C", 1.0, A, B).`



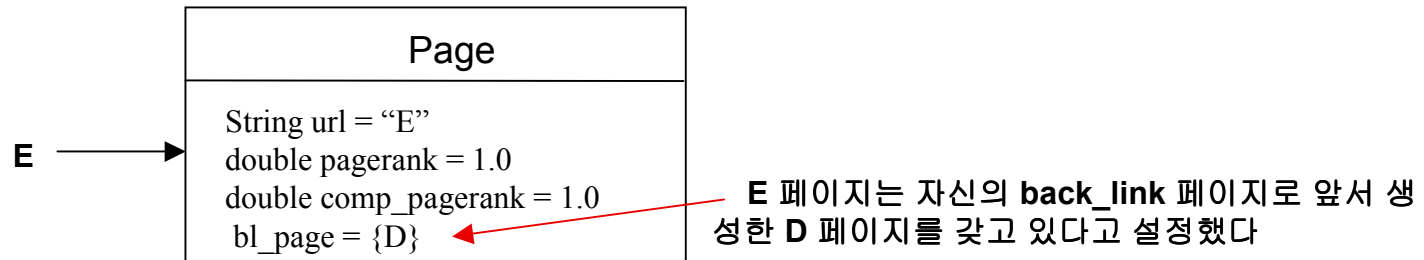
따라서 현재까지 아래와 같은 구조가 만들어졌다.



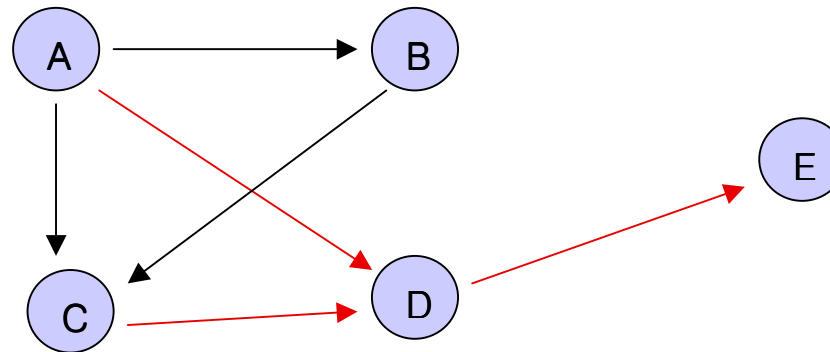
5) D 페이지를 만든다. `Page D = new Page("D", 1.0, C, A).`



6) E 페이지를 만든다. `Page E = new Page("E", 1.0, D)`

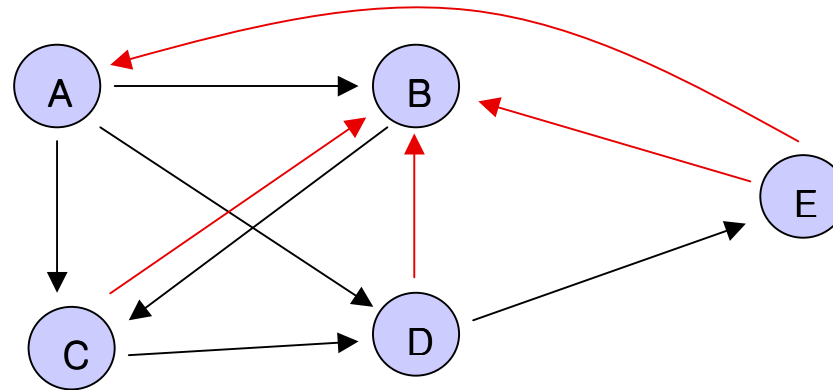


따라서 현재까지 아래와 같은 구조가 만들어졌다.

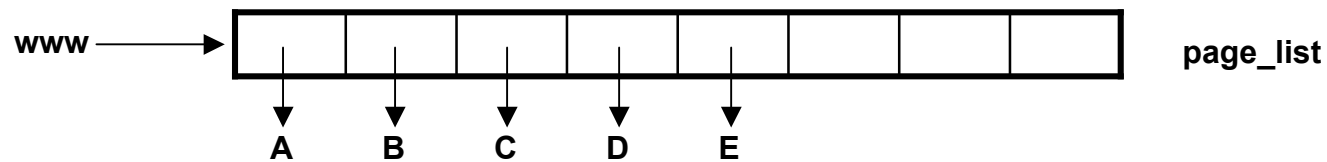


- 7) `A.insert_bl(E)`
- 8) `B.insert_bl(C, D, E)` `insert_bl` 메소드는 현 페이지에 새로운 `back_link` 페이지들을 설정한다.

따라서 아래와 같은 웹의 형태가 완성되었다



- 9) `www.insert_pages_build_web(A, B, C, D, E)` 위의 그림을 보면 사실 `Web` 클래스는 꼭 필요한 것이 아니라는 것을 알 수 있다. 하지만 프로그램 모양을 그럴듯하게 만들어 주어 `Main` 에서 실행을 쉽게 해 준다. 이 명령어가 실행되어 `www` 객체에 위의 페이지들이 들어가 웹이 만들어졌고, 이 웹페이지들에 대해서 페이지랭크를 계산할 준비가 다 되었다.



- 10) `www.display_webinfo();` Web 클래스에서 정의된 `display_webinfo( )` 메소드는 `page_list`에 있는 모든 페이지들에 대해서 그 페이지들의 정보를 출력하는 `display_pageinfo()` 메소드를 호출한다. 그 결과로 다음을 출력시켜 생각과 맞는 지 확인.

```
url= A : C= 3 : no_of_backlinks=1: comp_pagerank= 100.0 : pagerank_diff= 0.0
url= B : C= 1 : no_of_backlinks=4: comp_pagerank= -991.0 : pagerank_diff= 0.0
url= C : C= 2 : no_of_backlinks=2: comp_pagerank= 1.0 : pagerank_diff= 0.0
url= D : C= 2 : no_of_backlinks=2: comp_pagerank= 1.0 : pagerank_diff= 0.0
url= E : C= 2 : no_of_backlinks=1: comp_pagerank= 1.0 : pagerank_diff= 0.0
```

- 11) `System.out.println("");` 예쁘게 보이기 위해 한줄 띄고.

- 12-13) `int iterations = 200;`  
`for (int i=0 ; i<iterations ; i++)` for 루프내에 있는 페이지랭크 순환계산식을 200번 돌려본다

- 14) `www.calc_pagerank();` Web 클래스의 `calc_pagerank()` 메소드는 `page_list`에 있는 모든 페이지들에 대해서 Page 클래스에서 정의된 `calc_pagerank()` 메소드를 실행시킨다. Page 클래스의 `calc_pagerank()` 는 현 페이지의 `back_link` 페이지들의 C 와 페이지랭크값을 갖고 현 페이지의 페이지랭크를 아래의 식으로 계산한다.

$$PR(A) = (1-d) + d \cdot [PR(T1)/C(T1) + \dots + PR(Tn)/C(Tn)] \quad // T1 \dots Tn \text{ 들은 A의 back\_link}$$

- 15-16) `System.out.print(" " + (i+1) + " : " );` 지금 몇번째 루프를 돌고 있는가 표시하고, 연이어  
`www.display_PageRanks();` 각 페이지들의 계산된 현재 페이지랭크값을 표시한다. 페이지들의 페이지랭크값은 `page_list` 에 페이지들이 원래 삽입된 순서, 즉 A, B, C, D, E 페이지순으로 아래와 같이 출력된다.

횟수	A	B	C	D	E
1 :	0.575000	29.758333	-813.866667	28.908333	0.575000
2 :	0.394375	-333.050000	25.607500	-345.580417	12.436042
3 :	5.435318	-130.441432	-282.830760	11.144927	-146.721677

- 17) `www.update_pr_values();` 이번에 계산된 페이지랭크값들을 `update` 해서 다음번의 페이지랭크를 계산할 때 쓰도록 한다.
- 18-19) `if (i < iterations - 1)`  
`www.update_pr_values();` 이것은 `for` 루프의 마지막 회 때 페이지랭크값을 `update` 하지 않도록하여, 마지막에 계산된 페이지랭크값과 그 바로 전에 계산되었던 페이지랭크값의 차이를 비교하도록 한 것이다. 이 차이를 보고 페이지들의 페이지랭크값이 충분히 수렴되었다고 판단할 수 있다. 앞의 17) 번의 `statement`와 이 `statement` 중 하나를 써야한다.
- 20) `System.out.println("");` 한 줄 띄우고
- 21) `www.display_webinfo();` 현재 각 페이지들의 상태 (`url`명, `C` 값, `back_link` 페이지 수, 페이지랭크값, 바로전의 페이지랭크값과의 차이) 를 출력한다.