

자료구조 실습 보고서

숙제명: [제07주]이중 연결 링크드리스트 큐 구현

제출일: 2018.04.20

학번/이름: 201404377 / 진승언

<프로그램 설명서>

이번 과제는 이중 연결 링크드리스트로 큐를 구현하는 것이었다. head로 처음 노드를 가리키고 노드들은 prev와 next로 이중으로 연결되었다. 그리고 처음 노드의 prev는 마지막노드를 가리키고 있고 마지막 노드의 next는 첫 노드를 가리키고 있다.(마지막 노드의 next가 head를 가리키는게 아니라는 것을 주의해야한다.)

LinkedList 클래스의 add와 remove 메소드를 구현하는게 이번 과제의 주 목표였다. 먼저 add 메소드는 추가할 노드를 하나 생성해주고 만약 큐가 비어 있을 경우는 head와만 연결시킨다. 그리고 노드가 1개 들어있을 경우는 첫 노드와 이중연결을 시켜준다. 그 밖의 경우는(더 많은 노드가 있을 경우) 새로 추가하는 노드를 첫노드와 마지막노드와 이중으로 연결해주었다.

remove 메소드의 경우는 먼저 제거할 데이터를 반환시켜주어야 하니깐 peek()을 이용해서 반환할 데이터를 저장해둔다. 그다음 사이즈가 1일 경우는 헤드노드와 첫 노드밖에 없는 것이므로 둘의 연결을 그냥 끊어주면 된다. 그 밖의 경우는 제거할 첫 노드와 연결 되있는 두 번째 노드와 마지막노드의 연결을 끊어주게 코딩을 하면 된다.

그리고 LinkedList를 동작하게 하는 TestLinkedList 클래스는 이때까지 해왔던 과제처럼 하면 되는데 그 중 5번 내용출력은 getHead() 메소드로 헤드 노드를 받아온다음에 getNext()로 해당 노드의 데이터를 출력해주면서 setNext로 출력한 노드의 다음노드로 이동하게 해주면서 for문으로 모든 데이터를 출력하게 해주었다.

큐의 FIFO구조를 이해하고 이중연결리스트로 구현 할 수 있었다.

<실행 결과 분석>

```
1. [Queue] 내용 추가
2. [Queue] 내용 N개 삭제
3. [Queue] 내용 하나 삭제
4. [Queue] 사이즈 출력
5. [Queue] 내용 출력
6. [Queue] 첫번째 값 출력
7. 종료
```

1

추가하고 싶은 내용

A

[A]가 추가되었습니다.

```
1. [Queue] 내용 추가
2. [Queue] 내용 N개 삭제
3. [Queue] 내용 하나 삭제
4. [Queue] 사이즈 출력
5. [Queue] 내용 출력
6. [Queue] 첫번째 값 출력
7. 종료
```

1

추가하고 싶은 내용

B

[B]가 추가되었습니다.

```
1. [Queue] 내용 추가
2. [Queue] 내용 N개 삭제
3. [Queue] 내용 하나 삭제
4. [Queue] 사이즈 출력
5. [Queue] 내용 출력
6. [Queue] 첫번째 값 출력
7. 종료
```

1

추가하고 싶은 내용

C

[C]가 추가되었습니다.

6

B

```
1. [Queue] 내용 추가
2. [Queue] 내용 N개 삭제
3. [Queue] 내용 하나 삭제
4. [Queue] 사이즈 출력
5. [Queue] 내용 출력
6. [Queue] 첫번째 값 출력
7. 종료
```

2

삭제하고 싶은 것수:

2

[B]가 삭제되었습니다

[C]가 삭제되었습니다

```
1. [Queue] 내용 추가
2. [Queue] 내용 N개 삭제
3. [Queue] 내용 하나 삭제
4. [Queue] 사이즈 출력
5. [Queue] 내용 출력
6. [Queue] 첫번째 값 출력
7. 종료
```

5

D

```
1. [Queue] 내용 추가
2. [Queue] 내용 N개 삭제
3. [Queue] 내용 하나 삭제
4. [Queue] 사이즈 출력
5. [Queue] 내용 출력
6. [Queue] 첫번째 값 출력
7. 종료
```

4

사이즈는1입니다

7. 종료

1

추가하고 싶은 내용

D

[D]가 추가되었습니다.

```
1. [Queue] 내용 추가
2. [Queue] 내용 N개 삭제
3. [Queue] 내용 하나 삭제
4. [Queue] 사이즈 출력
5. [Queue] 내용 출력
6. [Queue] 첫번째 값 출력
7. 종료
```

5

A B C D

```
1. [Queue] 내용 추가
2. [Queue] 내용 N개 삭제
3. [Queue] 내용 하나 삭제
4. [Queue] 사이즈 출력
5. [Queue] 내용 출력
6. [Queue] 첫번째 값 출력
7. 종료
```

3

[A]가 삭제되었습니다

```
1. [Queue] 내용 추가
2. [Queue] 내용 N개 삭제
3. [Queue] 내용 하나 삭제
4. [Queue] 사이즈 출력
5. [Queue] 내용 출력
6. [Queue] 첫번째 값 출력
7. 종료
```

5

B C D

1. [Queue] 내용 추가

FIFO구조로 올바르게 실행됨을 볼 수 있다.