

자료구조 실습 보고서

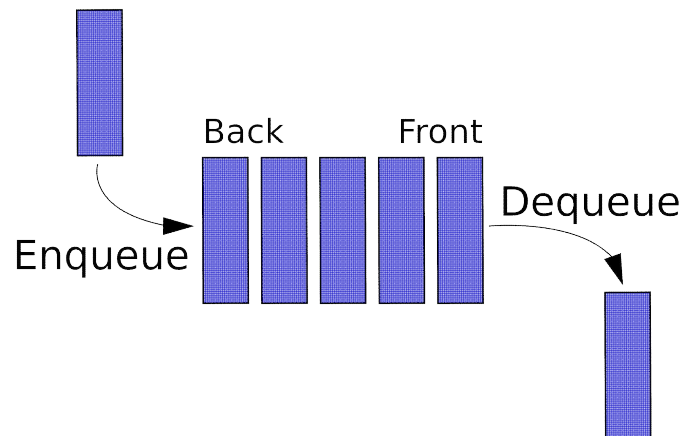
[제 7 주]

제출일 : 2018.4.19.목요일

학번 : 201604137

이름 : 김예지

[프로그램 설명서]



Queue: 나중에 집어 넣은 데이터가 먼저 나오는 스택과는 반대되는 개념이다. 큐는 컴퓨터의 기본적인 자료 구조의 한가지로, 먼저 집어 넣은 데이터가 먼저 나오는 FIFO (First In First Out)구조로 저장하는 형식을 말한다.

프린터의 출력 처리나 윈도우 시스템의 메시지 처리기, 프로세스 관리 등 데이터가 입력된 시간 순서대로 처리해야 할 필요가 있는 상황에 이용된다.

Linked Queue: 연결 리스트로 구현한 큐는 연결 리스트를 사용한 것으로써, 큐의 길이를 쉽게 늘릴 수 있어 오버플로우가 발생하지 않는 것이 특징이다.

필요에 따라 환형으로 만들 수도 있으며, 환형으로 만들지 않아도 삽입과 삭제가 제한되지 않아 편리하다.

[실행 결과 분석]

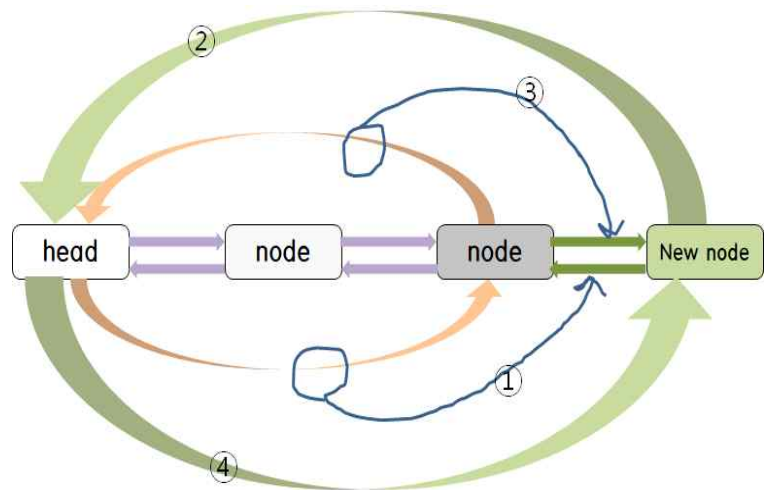
```
public class LinkedList {
    private Node head=new Node();
    private Node front;
    private Node rear;
    private int size;

    public LinkedList() {
        this.size=0;
        this.front=null;
        this.rear=null;
    }
}
```

front는 제일 첫 노드를 가리키며 삭제 할 때마다 다음 노드로 옮겨가며 가리킨다.
rear는 마지막 노드로 새로 노드를 추가할 때마다 추가하는 노드를 가리키며 옮겨간다.

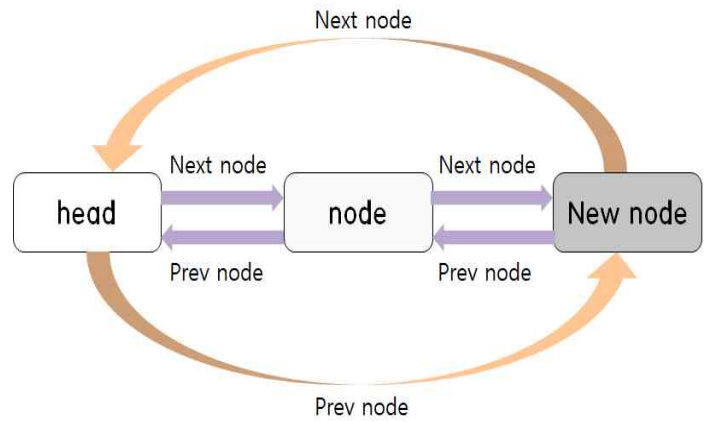
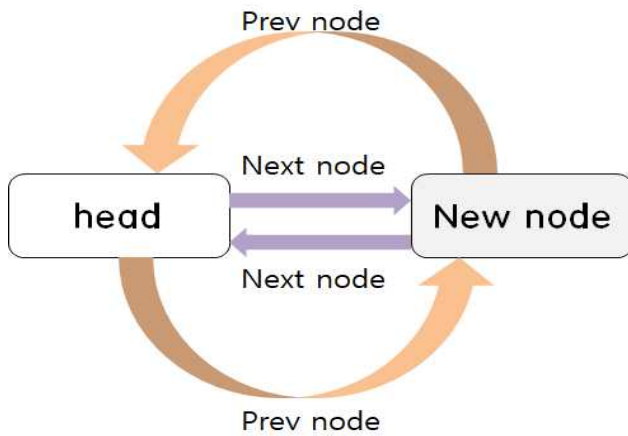
```
public void add(Object item) {
    Node newNode = new Node(item);

    if(isEmpty()) {
        rear=newNode;
        front=newNode;
        head.setNext(newNode);
        head.setPrev(newNode);
        newNode.setNext(head);
        newNode.setPrev(head);
    }else {
        newNode.setPrev(head.getPrev());
        newNode.setNext(head);
        head.getPrev().setNext(newNode);
        head.setPrev(newNode);
    }
    size++;
}
```



1.add: 현재 큐에 아무런 객체가 없다면 front, rear를 새로운 노드에 위치시키고 head의 다음을 새로운 노드로, 이전도 새로운 노드로 설정해준다. 새로운 노드도 마찬가지로

큐가 null이 아니라면 ①헤드가 가리키고 있던 prev를 새로운 노드의 prev로 재설정하고, ②새로운 노드의 next를 head로 세팅한다. 그리고 ③헤드의 prev의 다음노드(새로운 노드 입력 전 rear)를 새로운 노드가 되게끔 해주고 ④head의 prev를 새로운 노드로 세팅한다.

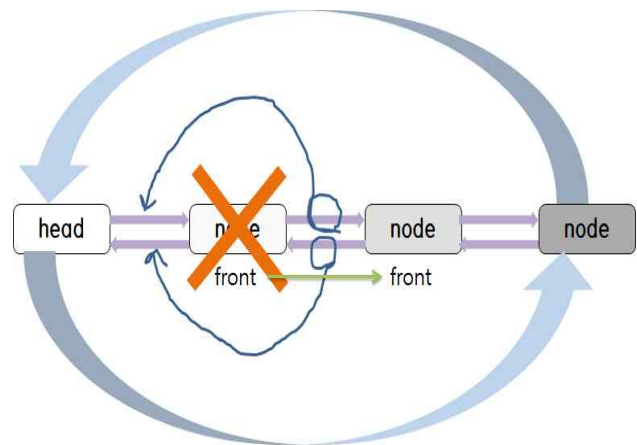


```

public Object remove() {
    Object item=peek();

    head.setNext(front.getNext());
    head.getNext().setPrev(head);
    front=front.getNext();
    size--;
    if(front==null) {
        rear=null;
    }
    return item;
}

```



2.remove: 큐는 먼저 넣었던 값부터 없어지므로 front의 값을 가져와 item 변수에 담고 head의 다음을 맨 앞 값의 다음 값으로 설정한다. 그리고 그 값의 prev를 head로 설정해주고 front는 한 칸 뒤의 값으로 설정해준다. 그리고 사이즈 1 감소시킨 후 item 반환.

```
//맨 앞에 있는 노드(front)를 꺼내줌
public Object peek() {
    if(isEmpty()) {
        throw new IllegalStateException("the queue is empty");
    }else {
        return front.getObj();
    }
}
```

peek: 큐에 값이 있다면 front가 현재 가리키고 있는 객체를 반환해준다.