



Queue<E> 컬렉션 클래스들

Queue<E> 컬렉션 클래스들

- 스택(Stack)과 큐(Queue)의 이해

- 스택(Stack)

- LIFO(last-in-first-out) : 가장 마지막에 저장된 데이터가 가장 처음 빠져나오는 자료구조
 - 아래가 막힌 긴 통에 비유할 수 있음

- 큐(Queue)

- FIFO(first-in-first-out) : 가장 먼저 저장된 데이터가 가장 처음 빠져나오는 자료구조
 - 입구와 출구가 따로 있는 긴 통에 비유할 수 있음

Queue<E> 컬렉션 클래스들

- Queue<E> 인터페이스와 큐(Queue)의 구현
 - Queue<E> 인터페이스를 대표하는 세 가지 메소드
 - boolean add(E e) : 넣기
 - E remove() : 꺼내기 (반환 및 삭제)
 - E element() : 확인하기 (반환)
 - 위 세 메소드는 꺼낼 인스턴스가 없을 때나 저장 공간이 부족할 때 예외 발생
 - 따라서 주로 아래 세가지 메소드 사용
 - boolean offer(E e) : 넣기, 넣을 공간이 부족하면 false 반환
 - E poll() : 꺼내기 (반환 및 삭제), 꺼낼 대상이 없으면 null 반환
 - E peek() : 확인하기 (반환), 확인할 대상이 없으면 null 반환

Queue<E> 컬렉션 클래스들

```
public class Person {  
    private String name;  
    private int age;  
    public Person(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }  
    @Override  
    public String toString() {  
        return name + " : " + age;  
    }  
}
```

Queue<E> 컬렉션 클래스들

● 스택(Stack)의 구현

- 자바에서 기본 자료구조 대부분을 지원하며 스택 자료 구조도 컬렉션 클래스 Stack<E>를 통해 지원
- 단 Stack<E>는 자바 초기에 정의된 클래스로 지금은 이전 코드와의 호환성 유지를 위해 존재하는 클래스
- 동기화된 클래스로 멀티 스레드에 안전하지만, 그만큼의 성능 저하 발생하여 대신 "덱(Deque)"라는 자료구조 사용
 - public interface Deque<E> extends Queue<E>
 - 외형 구조가 큐와 유사하지만 양쪽 끝에서 넣고 빼는 것이 가능하여 스택처럼 사용 가능
- Deque<E>의 대표 메소드
 - 앞으로 넣고, 꺼내고, 확인하기
 1. boolean offerFirst(E e) : 넣기, 공간이 부족하면 false 반환
 2. E pollFirst() : 꺼내기, 꺼낼 대상이 없으면 null 반환
 3. E peekFirst() : 확인하기, 확인할 대상이 없으면 null 반환
 - 뒤로 넣고, 꺼내고, 확인하기
 1. boolean offerLast(E e) : 넣기, 공간이 부족하면 false 반환
 2. E pollLast() : 꺼내기, 꺼낼 대상이 없으면 null 반환
 3. E peekLast() : 확인하기, 확인할 대상이 없으면 null 반환

Queue<E> 컬렉션 클래스들

```
import java.util.ArrayDeque;
import java.util.Deque;

public class ArrayDequeCollection {
    public static void main(String[] args) {
        Deque<String> deq = new ArrayDeque<>();
        // 앞으로 넣고
        deq.offerFirst("1.Box");
        deq.offerFirst("2.Toy");
        deq.offerFirst("3.Robot");
        // 앞에서 꺼내기
        System.out.println(deq.pollFirst());
        System.out.println(deq.pollFirst());
        System.out.println(deq.pollFirst());
    }
}
```

- 스택과 같은 동작을 위해 다음과 같이 쌍으로 이루어 메소드 호출
 - offerFirst & pollFirst : 앞으로 넣고 앞에서 꺼내기
 - offerLast & pollLast : 뒤로 넣고 뒤에서 꺼내기