



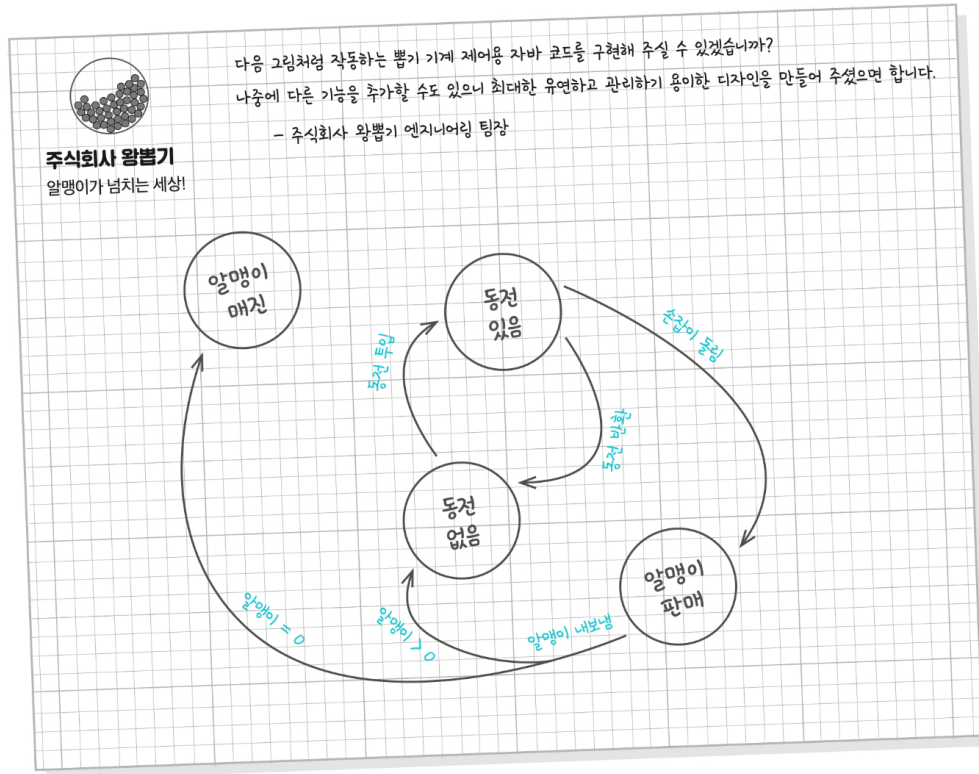
# 231010 상태 패턴

객체의 내부 상태가 바뀔에 따라서 객체의 행동을 바꿀 수 있다. 마치 객체의 클래스가 바뀌는 것과 같은 결과를 얻을 수 있다.

## Finite State Machine (FSM)

### 유한 상태 머신

- 유한 상태 기계는 자신이 취할 수 있는 유한한 갯수의 상태들을 가진다.
- 그리고 그 중에서 반드시 하나의 상태만 취한다.
- 현재 상태는 특정 조건이 되면 다른 상태로 변할 수 있다.
- 유한 상태 기계는 가능한 상태들의 집합과 각 상태들의 전이 조건으로 정의 될 수 있다.
- 상태들의 노드와 그 노드들을 연결하는 조건의 엣지로 표현할 수 있다(그래프).



## 유한 상태 기계를 왜 쓰는가?

- 가능한 상태들을 명확히 규정할 수 있다.
- 상태 중복을 피할 수 있다.
- 전이들을 명확하게 규정할 수 있다.
- 따라서 기계의 동작을 분명하게 규정할 수 있다.
- 프로그래밍에서 FSM에 기반한 객체를 만든다면, 안정적인 작동을 보장할 수 있다.

## 최첨단 뽑기 기계 만들기

- 현재 상태를 저장하는 인스턴스 변수 만들기 (enum)

'알맹이 매진' 상태를 나타냅니다.

```
final static int SOLD_OUT = 0;
final static int NO_QUARTER = 1;
final static int HAS_QUARTER = 2;
final static int SOLD = 3;
```

각 상태(동전 없음: NO\_QUARTER, 동전 있음: HAS\_QUARTER, 알맹이 판매: SOLD)마다 유일한 정수를 부여합니다.

```
int state = SOLD_OUT;
```

현재 상태를 저장하는 인스턴스 변수.

뽑기 기계를 상자에서 꺼내서 처음 설치하는 시점에서는 알맹이가 하나도 없을 테니까 일단 알맹이 매진 상태로 설정합니다.

## • 행동들을 모아 메소드로 만들기

```
public void insertQuarter() {
```

```
    if (state == HAS_QUARTER) {
```

조건문으로 상태를 확인합니다.

```
        System.out.println("동전은 한 개만 넣어주세요.");
```

```
    } else if (state == NO_QUARTER) {
```

그리고 가능한 상태별로 적절한 행동을 취합니다.

```
        state = HAS_QUARTER;
```

```
        System.out.println("동전이 투입되었습니다.");
```

```
    } else if (state == SOLD_OUT) {
```

하지만 다이어그램에서 봤던 것처럼 다른 상태로 넘어갈 수도 있습니다.

```
        System.out.println("매진되었습니다. 다음 기회에 이용해주세요.");
```

```
    } else if (state == SOLD) {
```

```
        System.out.println("알맹이를 내보내고 있습니다.");
```

```
    }
```

```
}
```

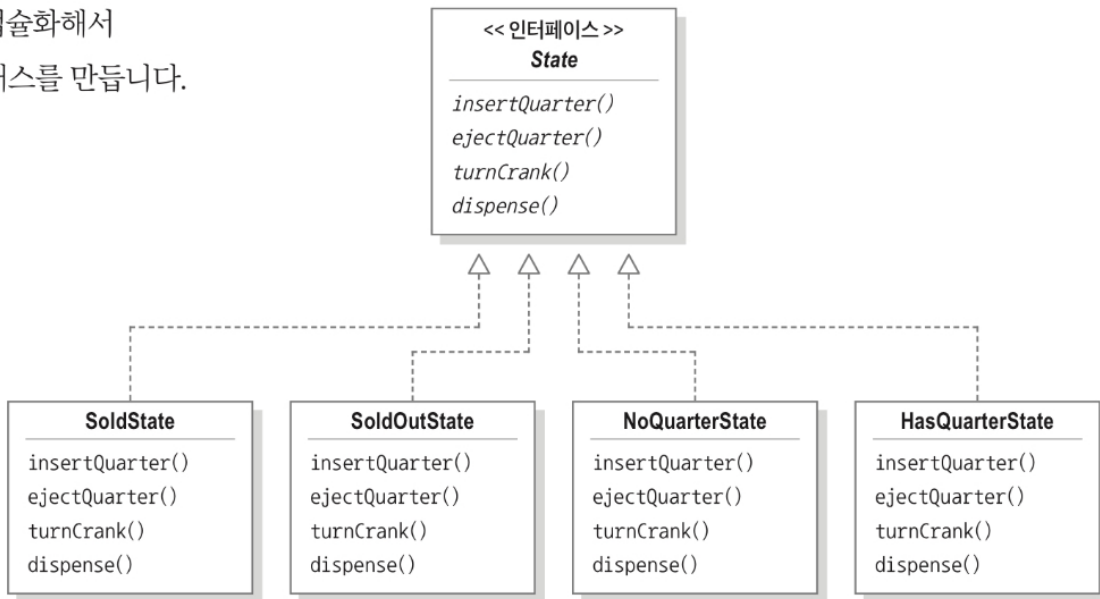
앗 일단 굴러가긴 굴러간다! (대신 if와 switch 떡칠이지만)

보너스 알맹이 기능을 추가해야할때..?

→ 각 메소드 별로 state == WINNER 를 검사하는 조건문을 일일이 추가해야함!

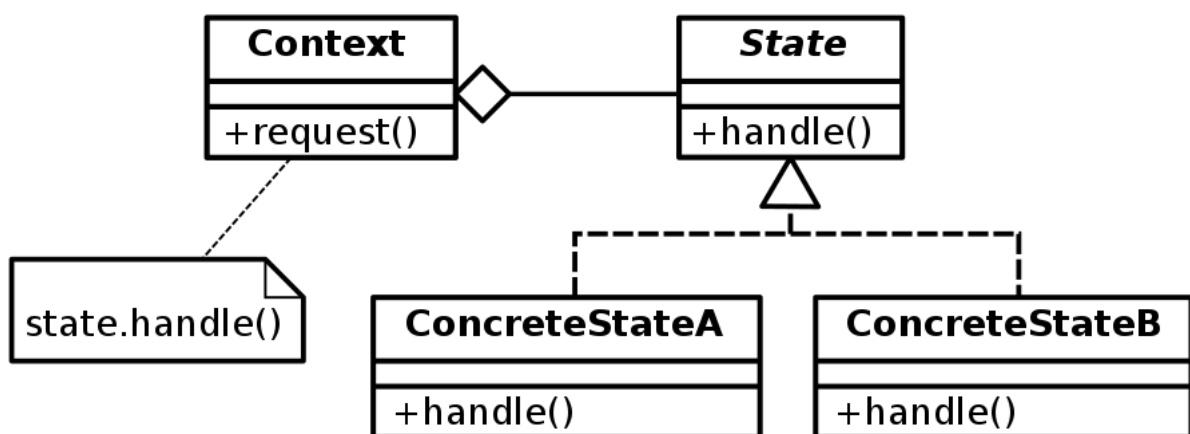
## 상태 패턴 적용해보기

· 캡슐화해서  
클래스를 만듭니다.



435p 뇌단련 어떻게 생각하시나요 ?  
사용하지 않는 메소드에 대한 처리..!

## 클래스 다이어그램



## 상태 패턴 VS 전략패턴

## 상태 패턴

- 인스턴스 생성 후 상태가 빈번하게 변경됨
- 상태 객체간의 존재 파악이 필요함
  - HasQuaterState에서 turnCrank()호출 후 WinnerState 나 SoldState로 변경해주는 단계 있음

## 전략 패턴

- 인스턴스 생성 후 상태가 거의 바뀌지 않음
  - 고무오리가... 청동오리가 되진 않잖아요
- 상태 객체간의 존재 파악이 필요없음
  - 고무오리는 청동오리에 간섭하지 않음