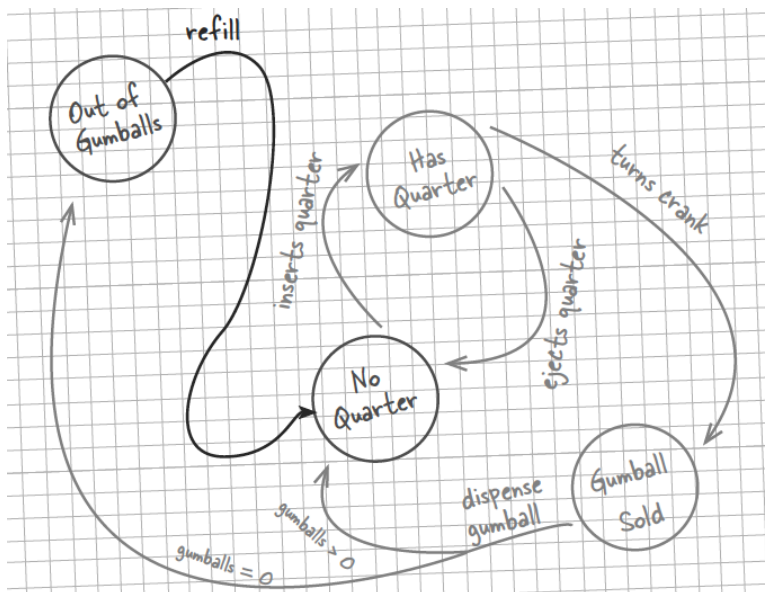


소프트웨어 설계 패턴 (실습 11)

이번 실습에서는 예제 코드로 주어진 gumballstatewinner 프로젝트를 바탕으로 진행합니다.

- 아래 그림과 같이 리필(refill)하는 부분을 상태 다이어그램에 추가하였는데, 이 부분을 다음과 같이 구현하시오.
 - GumballMachine에서 입력된 gumballs 개수 만큼 GumballMachine의 count 변수를 증가시키는 refill(int gumballs) 메소드를 추가하시오.
 - 리필 액션에 따른 상태 변화를 구현하기 위해, State 인터페이스와 각 상태 클래스들을 수정하시오.
 - 기존 코드에서 어느 부분이 변경되었는지를 기술하고, `GumballMachineTestDrive`에 이를 테스트할 수 있는 코드를 추가하여 실행결과를 제출하시오.



- 현재, 상태 클래스들은 중복된 코드 부분이 상당히 많이 있습니다.
 - State 인터페이스를 추상클래스로 만들고, 몇 가지 공통 기본 기능을 추상클래스에 정의함으로써, 상태 클래스들 사이의 코드의 중복을 해결해 보세요.

B. 어느 부분을 어떻게 수정하였는지, 어떻게 좋아 졌는지 서술하세요.

3. 현재 상태 전이는 각 상태 클래스에서 담당합니다. 상태 전이를 GumballMachine 에서 담당하도록 코드를 수정해 보세요. 그리고, 이러한 방식의 장점과 단점에 대해서 논의하세요