

우선 이 게임의 핵심인 하노이 탑 그 자체만을 추상하여 Tower 클래스를 만들었다.

1차 배열의 인덱스가 블록의 크기, 깊이 블록의 위치(FROM or TEMP or TO)를 표현하는 구조다.

2차 배열로 탑을 구현하는 방법도 생각했지만 화면에 출력할 때만 유용한 것 같아 기각했다.

지금 생각해보면 역시 잘한 선택이었다.

Tower 클래스 멤버 변수론 블록의 개수인 num, 블록들의 정보를 가지는 block가 전부다.

또, 멤버 함수로는 타워를 움직이는 move, 해당 위치의 최소 블록 크기의 인덱스를 리턴하는 getMin이 전부다. getMin은 move를 위한 조건을 판단하기 위해 만들어졌다.

한 블록을 A에서 B로 옮길 때, A에 블록이 있는지 그리고 A의 꼭대기(최소 블록)가 B의 꼭대기보다 작은가를 판단하여 옮기는 것이 핵심 원리다.

다음은 UI 클래스, 이 게임의 인터페이스다.

설계한 게임의 구조는 이렇다.

```
Tower *tower;
int attempt;

do {
    initialization();
    display();
    do {
        moveBlock();
        display();
    } while (!isFinish());
    celebration();
} while (isOneMore());
```

tower : 게임에 사용될 Tower 포인터

attempt : 블록을 옮긴 횟수

initialization : 사용자에게 블록 크기를 입력 받아 타워를 만들고 attempt를 초기화한다.

display : 현재 블록들을 화면에 표시한다.

moveBlock : 사용자에게 블록을 옮기는 인터페이스를 제공한다.

isFinish : 조건이 만족되었는지 검사한다. (블록이 전부 TO에 있는가?)

celebration : attempt에 따라 결과를 달리하여 보여준다. 타워를 딜리트한다.

isOneMore : 사용자에게 종료 의사를 묻는다.

UI 멤버의 세부 코드 설명은 별로 중요하지 않아 생략하겠다.

객체 지향적 게임을 개발하는 것이 목표였는데 어느 정도 달성한 것 같아 기분이 좋다.

왜 자바의 main 함수가 클래스 안에 있는지 알 것도 같다.

main 함수를 UI의 멤버 함수로 만들면 완전한 객체 지향 프로그램이 될 것이다.