

## Master-Slave Pattern

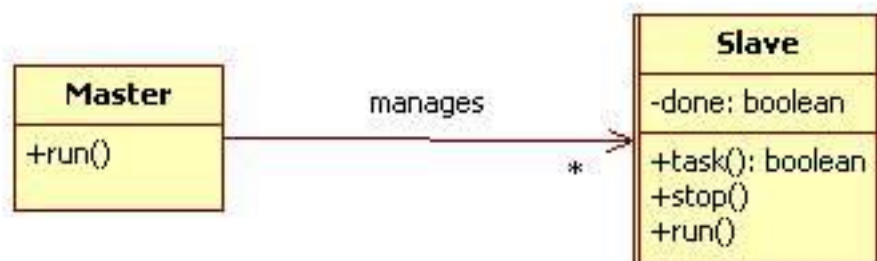
### 1. Master-Slave 패턴 [1]

문제의 분할-정복(Divide-and-Conquer) 원칙에 따라 작업을 나누기 위해 사용한다.

- Master: 여러 Slave 컴포넌트들에게 작업을 분할 후, Slave가 작업을 마치면 반환한 결과를 계산한다.
- Slave: Master를 대신해 작업할 서버서비스를 구현한다.

### 2. Master-Slave 패턴 구조 [1]

Master와 Slave 모두 Active Object로 각각이 실행 가능한 단위여야 한다. Master는 실행 단위로서 Slave가 되는 실행 단위를 생성하고 시작할 책임이 있다.



### 3. Master-Slave 패턴 사용

코드 템플릿에 있는 Master-Slave 패턴을 사용하는 메인 함수는 다음과 같다.

```

public class TestMaster {
    public static void main(String[] args) {
        Master master = new Master();
        master.run();
    }
}
  
```

### 4. Master-Slave 패턴 활용

- (1) 코드 템플릿의 Master는 10개의 Slave를 만든다. 1부터 100까지를 각각의 Slave에 할당하여 전체 Sum을 출력하도록 수정해 보자.

### 5. References

- [1] Master-Slave Pattern - <http://www.cs.sjsu.edu/~pearce/modules/lectures/oop/templates/threads/index.htm>