

# Quiz. SoundableExample.java (class)

SoundableExample 클래스는 아래와 같다. Soundable 인터페이스는 객체의 소리를 리턴하는 `sound()` 추상메서드를 갖습니다. SoundableExample의 `printSound()`는 Soundable 인터페이스 타입의 매개변수를 갖습니다. 아래와 같이 호출되도록 Soundable 인터페이스, Dog, Cat 클래스를 생성하시오.

```
public class SoundableExample {  
    private static void printSound(Soundable soundable) {  
        System.out.println(soundable.sound());  
    }  
    public static void main(String[] args) {  
        printSound(new Cat());  
        printSound(new Dog());  
    }  
}
```

출력결과

야옹  
멍멍

# Quiz. DaoExample.java (class)

DaoExample 클래스는 아래와 같다. dbWork() 메서드는 OracleDao와 MySqlDao 객체를 매개값으로 주고 호출했습니다. dbWork() 메서드는 두 객체를 모두 매개값으로 받기 위해 DataAccessObject 타입의 매개변수를 가지고 있습니다. 결과를 보고 DataAccessObject 인터페이스와 OracleDao와 MySqlDao 구현 클래스를 작성하세요.

```
public class DaoExample {  
    public static void dbWork(DataAccessObject dao)  
    {  
        dao.select();  
        dao.insert();  
        dao.update();  
        dao.delete();  
    }  
    public static void main(String[] args) {  
        dbWork(new OracleDao());  
        dbWork(new MySqlDao());  
    }  
}
```

## 출력결과

```
Oracle DB에서 검색  
Oracle DB에 삽입  
Oracle DB를 수정  
Oracle DB에서 삭제  
MySql DB에서 검색  
MySql DB에 삽입  
MySql DB를 수정  
MySql DB에서 삭제
```

# Quiz. Printer.java (class)

프린트 출력 기능을 만들기 위한 인터페이스를 만든다.

```
public interface Printer {  
    void print(String doc);  
}
```

삼성프린트와 LG프린트는 모두 프린트출력 기능을 가지고 있기 때문에 위 인터페이스를 구현받는다.

Ex클래스에서 각각의 프린트를 생성하고 인쇄를 진행한다.

삼성 프린터  
**자바 이론 문서**를 인쇄합니다.  
LG 프린터  
**자바 이론 문서**를 인쇄합니다.

# Quiz. Printer.java (class)

앞의 예제에서 삼성프린터가 세계 최초 컬러 프린터 기능(interface ColorPrintabl)을 추가했다.

인터페이스 상속을 이용하여 기능을 추가하고 아래와 같이 호출하시요.

삼성 프린터  
자바 이론 문서를 인쇄합니다.  
삼성 프린터  
컬러로자바 이론 문서를 인쇄합니다.  
LG 프린터  
자바 이론 문서를 인쇄합니다.

# Quiz. SchedulerTest.java (class)

고객 센터에 전화가 오면 대기열에 저장된다. 상담원이 지정되기 전까지 대기상태가 된다. 각 전화를 상담원에게 배분하는 정책은 여러 방식으로 구현할 수 있다.

Interface(Scheduler- getNextCall(), sendCallToAgent())를 생성하고 구현클래스를 아래와 같이 생성하시오. (단순출력)

## 1. 순서대로 배분(RoundRobin):

getNextCall() 상담 전화를 순서대로 대기열에서 가져옵니다.

sendCallToAgent()다음 순서 상담원에게 배분합니다.

## 2. 짧은 대기열 찾아 배분(LeastJob): 상담 전화를 순서대로 대기열에서 가져옵니다. 현재 상담업무가 없거나 상담대기가 가장 적은 상담원에게 할당합니다.

## 3. 우선순위에 따라 배분(PriorityAllocation): 고객의 등급이 높은 고객의 전화를 먼저 가져옵니다.

업무 skill 이 가장 높은 상담원의 대기열에 앞에 우선 배분합니다.

전화 상담 배분방식을 선택하세요

R : 한명씩 차례로 할당

L : 쉬고 있거나 대기가 가장 적은 상담원에게 할당

P : 우선순위가 높은 고객 먼저 할당

r

상담 전화를 순서대로 대기열에서 가져옵니다.

다음 순서 상담원에게 배분합니다.

R or r

상담 전화를 순서대로 대기열에서 가져옵니다.

다음 순서 상담원에게 배분합니다.

L or l

상담 전화를 순서대로 대기열에서 가져옵니다.

현재 상담업무가 없거나 상담대기가 가장 적은 상담원에게 할당합니다.

P or p

고객의 등급이 높은 고객의 전화를 먼저 가져옵니다.

업무 skill 이 가장 높은 상담원의 대기열에 앞에 우선 배분합니다.

## Quiz. 학사정보관리 시스템

### \* 회원 가입

회원 가입할 때

이름, 아이디, 비밀번호를 입력받고 선생은 담당 과목, 학생은 전공을 선택합니다.

### \*로그인

로그인 실패시 아이디 또는 비밀번호 실패 메시지가 나오게 합니다.

로그인 성공시

선생은 "강의를 합니다", "채점을 합니다" 출력

학생은 "공부를 합니다", "시험을 봅니다" 출력

\*프로그램 종료시 종료 메시지가 나오게 합니다.

선생과 학생 정보는 Member 클래스로 부터 상속받게 만듭니다.

단 선생과 학생의 기능을 task 인터페이스에 정의하고 Member클래스는 task 를 구현받은 추상클래스로 제작합니다.

1단계: task인터페이스에 work(), test()추상 메소드 생성

2단계 : task 를 구현받은 추상클래스 Member 구성

- 필드는 이름, 아이디, 비밀번호 ,position

- 생성자는 기본 필드를 초기화 하는 생성자로 구성

3단계 : Member클래스를 상속받은 Teacher클래스와 Student 클래스를 작성

- Teacher클래스는 추가로 담당 과목 필드를 추가하고 생성자로 초기화

- Student클래스는 추가로 전공 필드를 추가하고 생성자로 초기화

- 위 두 클래스 모두 추상 메소드를 오버라이딩

4단계 : 메인 메뉴를 구성

회원 가입, 로그인, 프로그램 종료

5단계 : 각각의 기능을 작성해 봅니다.

학사 정보 관리 시스템  
메뉴를 선택하세요.

1. 회원 가입
2. 로그인
3. 프로그램 종료

1

회원 가입을 진행합니다.  
ID를 입력하세요.

lee  
암호를 입력하세요.

1234  
이름을 입력하세요.

je  
1. 선생님 | 2. 학생

1

담당 과목을 입력하세요  
Com

학사 정보 관리 시스템  
메뉴를 선택하세요.

1. 회원 가입
2. 로그인
3. 프로그램 종료

2

로그인 절차를 진행합니다.

ID를 입력하세요.

lee  
암호를 입력하세요.

1234  
강의를 합니다.  
채점을 합니다.

학사 정보 관리 시스템  
메뉴를 선택하세요.

1. 회원 가입
2. 로그인
3. 프로그램 종료

1

회원 가입을 진행합니다.  
ID를 입력하세요.

hong  
암호를 입력하세요.

1234  
이름을 입력하세요.

jd  
1. 선생님 | 2. 학생

2

전공을 입력하세요  
Com

학사 정보 관리 시스템  
메뉴를 선택하세요.

1. 회원 가입
2. 로그인
3. 프로그램 종료

2

로그인 절차를 진행합니다.  
ID를 입력하세요.

hong  
암호를 입력하세요.

1234  
공부를 합니다.  
시험을 봅니다.

학사 정보 관리 시스템  
메뉴를 선택하세요.

1. 회원 가입
2. 로그인
3. 프로그램 종료

3

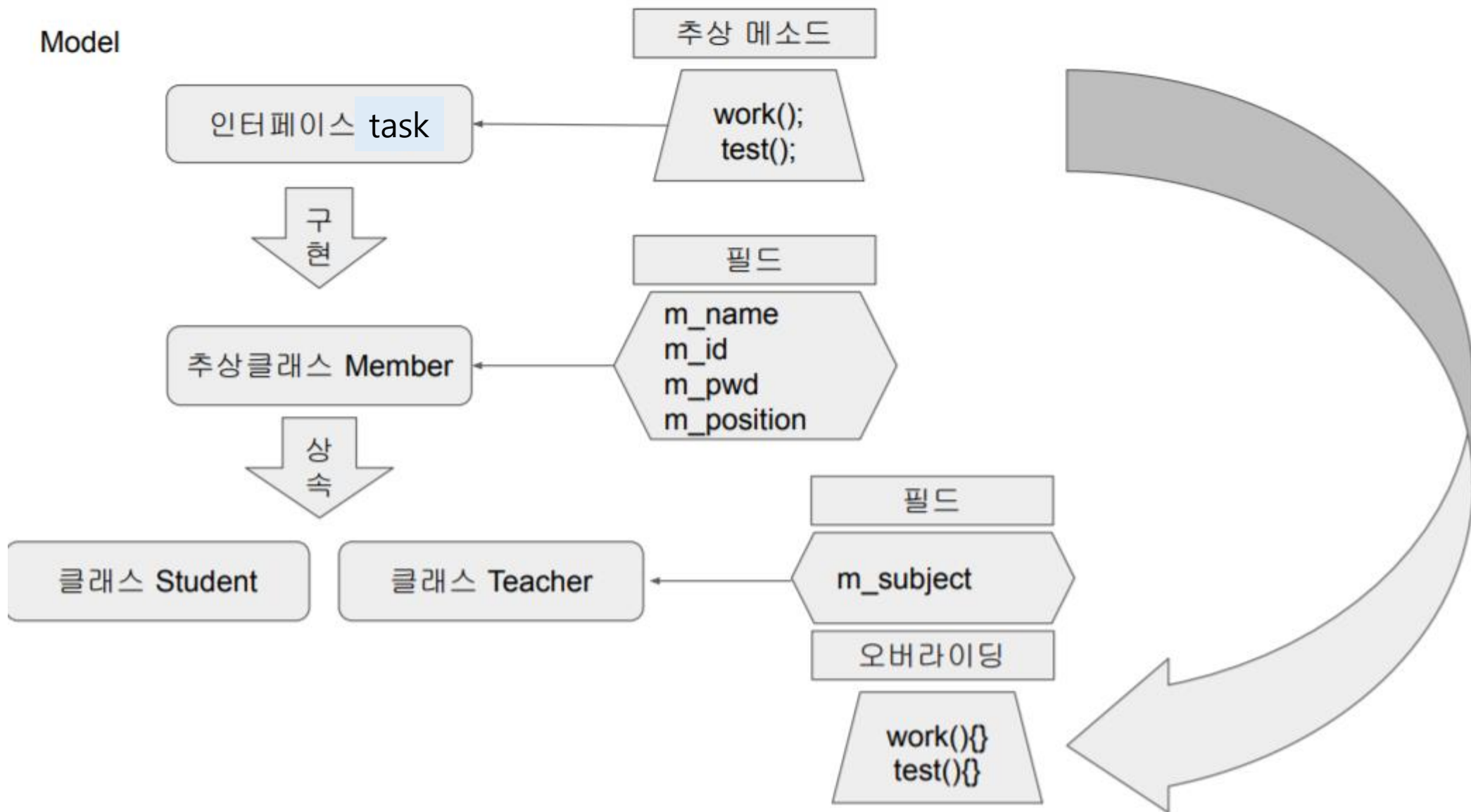
프로그램을 종료하시겠습니까?  
(y/n)

y

프로그램을 종료합니다.

# 학사정보관리 시스템

Model





# 학사정보관리 시스템

