



JAVA 객체지향 프로그래밍 & DESIGN PATTERN (연습문제)

문성훈

Copyright 2017 에이티지랩

➤ 문제

● Class 명 : Account

- 계좌를 관리하기 위해서 클래스가 필요하며, Account라는 이름의 class를 만든다.
- 필요로 하는 정보로는 예금주, 계좌번호, 잔액이 있고, 입금(deposit)과 출금(withdraw)할 수 있고, 현 잔액에 대한 조회가 가능하다.

- Account class를 이용하여 다음과 같은 실행 결과가 나오도록 필드에 값을 저장하고, 적절한 method를 이용하여 출력한다.

➤ 실행 결과

계좌 123-456789 (예금주 : 홍길동)
잔액 : 10000 원
20000 원 입금합니다.
잔액 : 30000 원
45000 원 출금합니다.
잔액 : -15000 원

➤ 문제

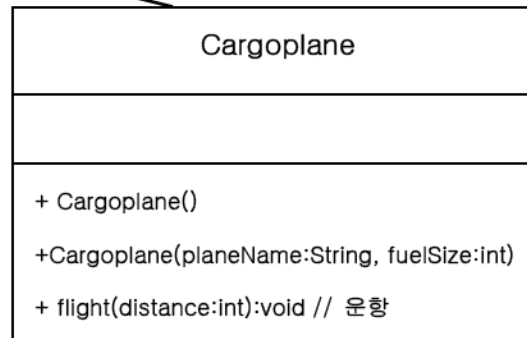
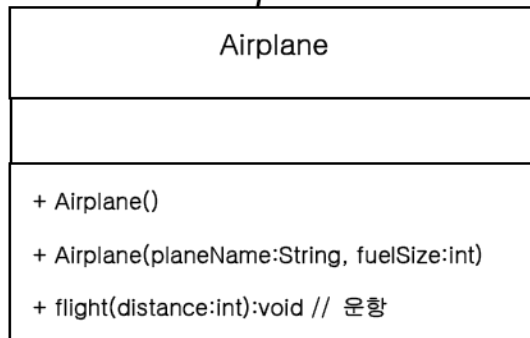
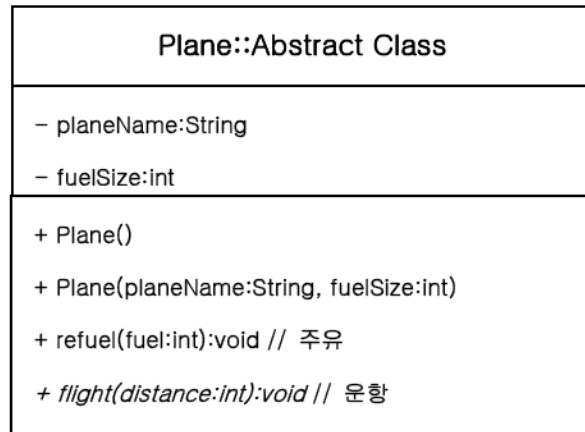
- Random class 이용하여 임의의 정수를 10개 생성한 후 배열에 저장하고 그 배열을 return하는 method를 작성한 후 해당 배열을 console에 출력하세요.
- class명과 method 명은 임의로 작성합니다.
- 임의의 정수는 1부터 30까지의 수를 random하게 발생시킵니다.
- 단, 중복된 숫자가 발생하지 않도록 프로그램을 작성합니다.

➤ 문제

Abstract 클래스에서 상속받은 두 개의 클래스를 구현하여 실행 결과의 예와 같이 나오도록 프로그램을 작성한다.

1. 사용 데이터

| class Type | planeName | fuelSize |
|------------|-----------|----------|
| Airplane | L747 | 1000 |
| Cargoplane | C40 | 1000 |



➤ 문제 (cont.)

| Package명 | 클래스명 | 메소드 | 설명 |
|-----------|------------|---|--|
| planetest | Plane | +Plane() | 기본 생성자 |
| | | +Plane(planeName:String, fuelSize:int) | 2개의 클래스 변수를 받는 생성자 |
| | | +refuel(fuel:int):void | 일정 양의 연료 주입, 기존 연료가 증가됨 |
| | | +flight(distance:int):void | 일정 거리 만큼 운항, 연료 감소 추상 함수 하위 객체에서 반드시 구현해야 함 |
| | Airplane | +Airplane() | 기본 생성자 |
| | | +Airplane(planeName:String, fuelSize:int) | 2개의 클래스 변수를 받는 생성자 |
| | | +flight(distance:int):void | 10 운항 시 연료 30감소 |
| | Cargoplane | +Cargoplane() | 기본 생성자 |
| | | +Cargoplane(planeName:String, fuelSize:int) | 2개의 클래스 변수를 받는 생성자 |
| | | +flight(distance:int):void | 10운항 시 연료 50감소 |
| | PlaneTest | + main(args:String[]):void | main 함수를 이용 하여 동작 |

➤ 문제 (cont.)

4. PlaneTest 클래스 구조

```
public class PlaneTest {  
  
    public static void main(String args[]) {  
        // Airplane과 Cargoplane 객체 생성  
        // 생성된 객체의 정보 출력  
        // Airplane과 Cargoplane 객체에 100씩 운항  
        // 운항 후 객체의 변경된 정보 출력  
        // Airplane과 Cargoplane 객체에 200씩 주유  
        // 주유 후 객체의 변경된 정보 출력  
    }  
}
```

5. 실행 결과 예

| Plane | fuelSize |
|-------|----------|
| L747 | 1000 |
| C40 | 1000 |

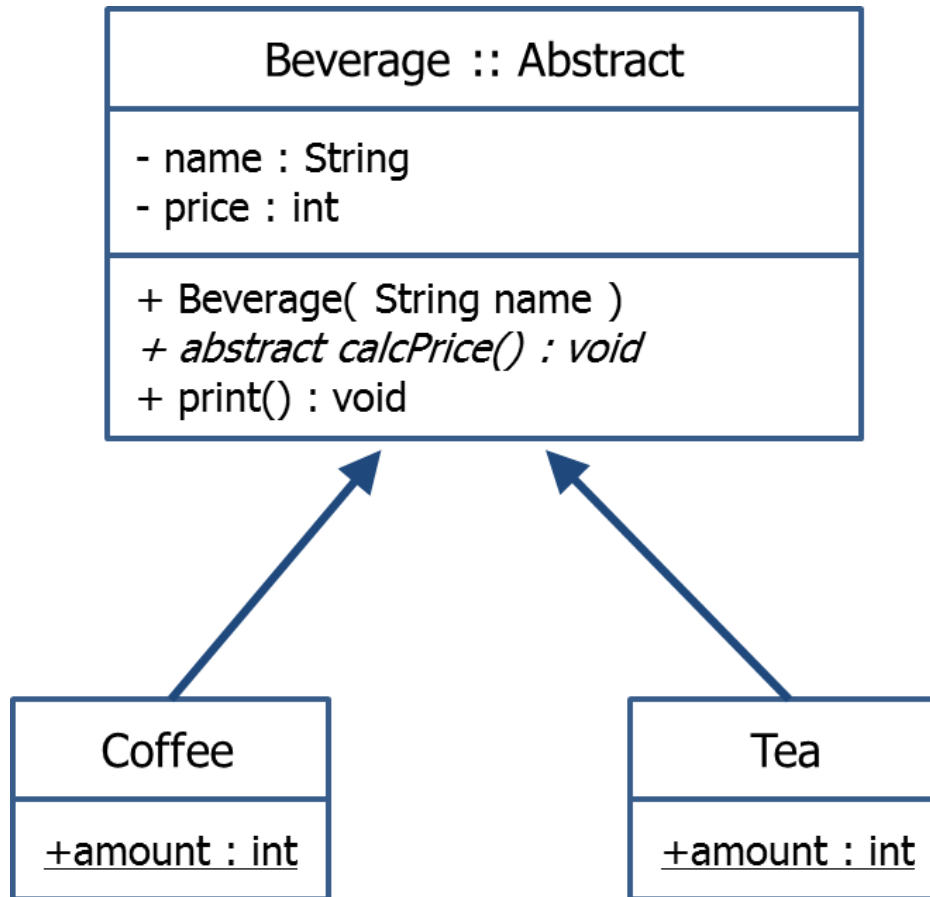
100 운항

| Plane | fuelSize |
|-------|----------|
| L747 | 700 |
| C40 | 500 |

200 주유

| Plane | fuelSize |
|-------|----------|
| L747 | 900 |
| C40 | 700 |

➤ 문제



➤ 문제 (cont.)

● 이전 장의 class diagram을 본 후

- 각각의 class를 작성합니다.
- Beverage는 abstract method를 가지는 abstract class입니다.
- Beverage의 print()는 다음의 내용을 화면에 출력하는 method입니다.

“음료는 000 이며, 가격은 000 입니다.”

- Coffee와 Tea class의 생성자는 적절히 작성
- 판매수량이 amount에 누적될 수 있도록 생성자에 작성.
- Coffee와 Tea class에서 calcPrice() method는 다음장의 조건을 적용합니다.

➤ 문제 (cont.)

- Coffee의 경우 다음과 같은 조건을 적용합니다.
 - Coffee name을 이용하여 price 설정
 - Americano : 1500
 - Cafe latte : 2500
 - Cappuccino : 3000
- Tea의 경우 다음과 같은 조건을 적용합니다.
 - Tea name을 이용하여 price 설정
 - Lemon Tea : 1500
 - Jasmine Tea : 2000
 - Black Tea : 2500
- 다음장의 결과가 나오도록 프로그램을 작성합니다.

➤ 문제 (cont.)

***** Coffee Shop 영업 개시 *****

1 번째 판매 음료 : Cappuccino, 가격은 3000

2 번째 판매 음료 : Cafe latte, 가격은 2500

3 번째 판매 음료 : Jasmine Tea, 가격은 2000

4 번째 판매 음료 : Cafe latte, 가격은 2500

5 번째 판매 음료 : Black Tea, 가격은 2500

총 판매금액 : 12,500

Coffee 판매 개수 : 3잔

Tea 판매 개수 : 2잔



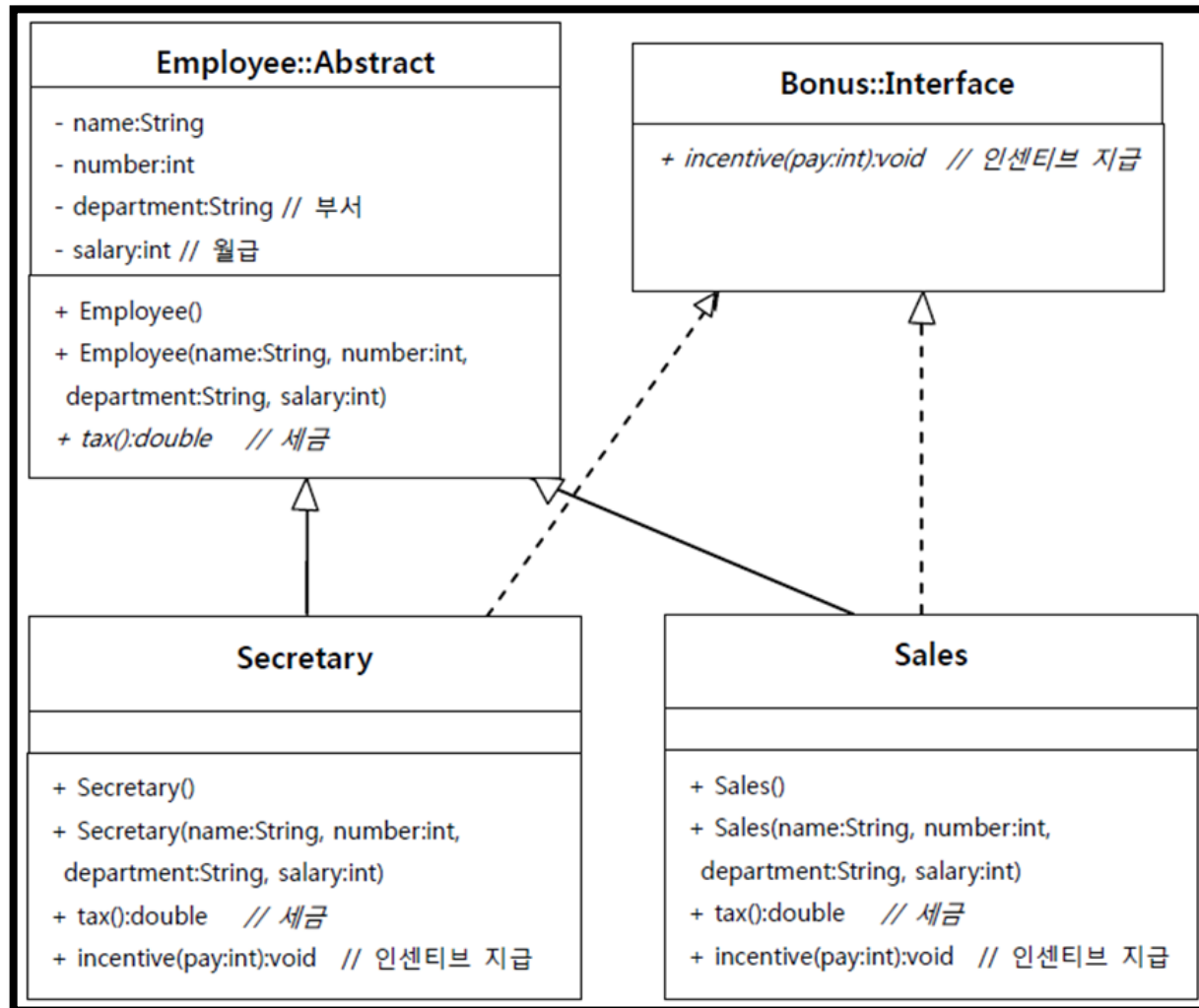
➤ 문제

Abstract 클래스를 상속받고 Interface를 implements한 두 개의 클래스를 구현하여 실행 결과의 예와 같이 나오도록 프로그램을 작성한다.

1. 사용 데이터

| name | number | department | salary |
|---------|--------|------------|--------|
| Hilery | 1 | secretary | 800 |
| Clinton | 2 | sales | 1200 |

➤ 문제 (cont.)



➤ 문제 (cont.)

| Package명 | 클래스명 | 메소드 | 설명 |
|----------|-----------|---|---|
| | Employee | +Employee() | 기본 생성자 |
| | | +Employee(name:String, number:int, department:String, salary:int) | 4개의 클래스 변수를 받는 생성자 |
| | | +tax():double | 세금을 리턴한다. |
| | Bonus | +incentive(pay:int):void | 인센티브를 지급한다. |
| | Secretary | + Secretary() | 기본 생성자 |
| | | + Secretary(name:String, number:int, department:String, salary:int) | 4개의 클래스 변수를 받는 생성자 |
| | | +tax():double | 세금을 리턴한다. salary에 10% 징수 |
| | | +incentive(pay:int):void | 인센티브 지급. 지급되는 pay에 80%가 기존 salary에 더해진다. |
| | Sales | + Sales() | 기본 생성자 |
| | | + Sales(name:String, number:int, department:String, salary:int) | 4개의 클래스 변수를 받는 생성자 |
| | | +tax():double | 세금을 리턴한다. salary에 13% 징수 |
| | | +incentive(pay:int):void | 인센티브 지급. 지급되는 pay에 120%가 기존 salary에 더해진다. |
| | Company | +main(args:String[]):void | main 함수를 이용하여 동작 |

➤ 문제 (cont.)

| name | department | salary |
|---------|------------|--------|
| Hilery | secretary | 800 |
| Clinton | sales | 1200 |

인센티브 100 지급

| name | department | salary | tax |
|---------|------------|--------|-------|
| Hilery | secretary | 880 | 88.0 |
| Clinton | sales | 1320 | 171.6 |

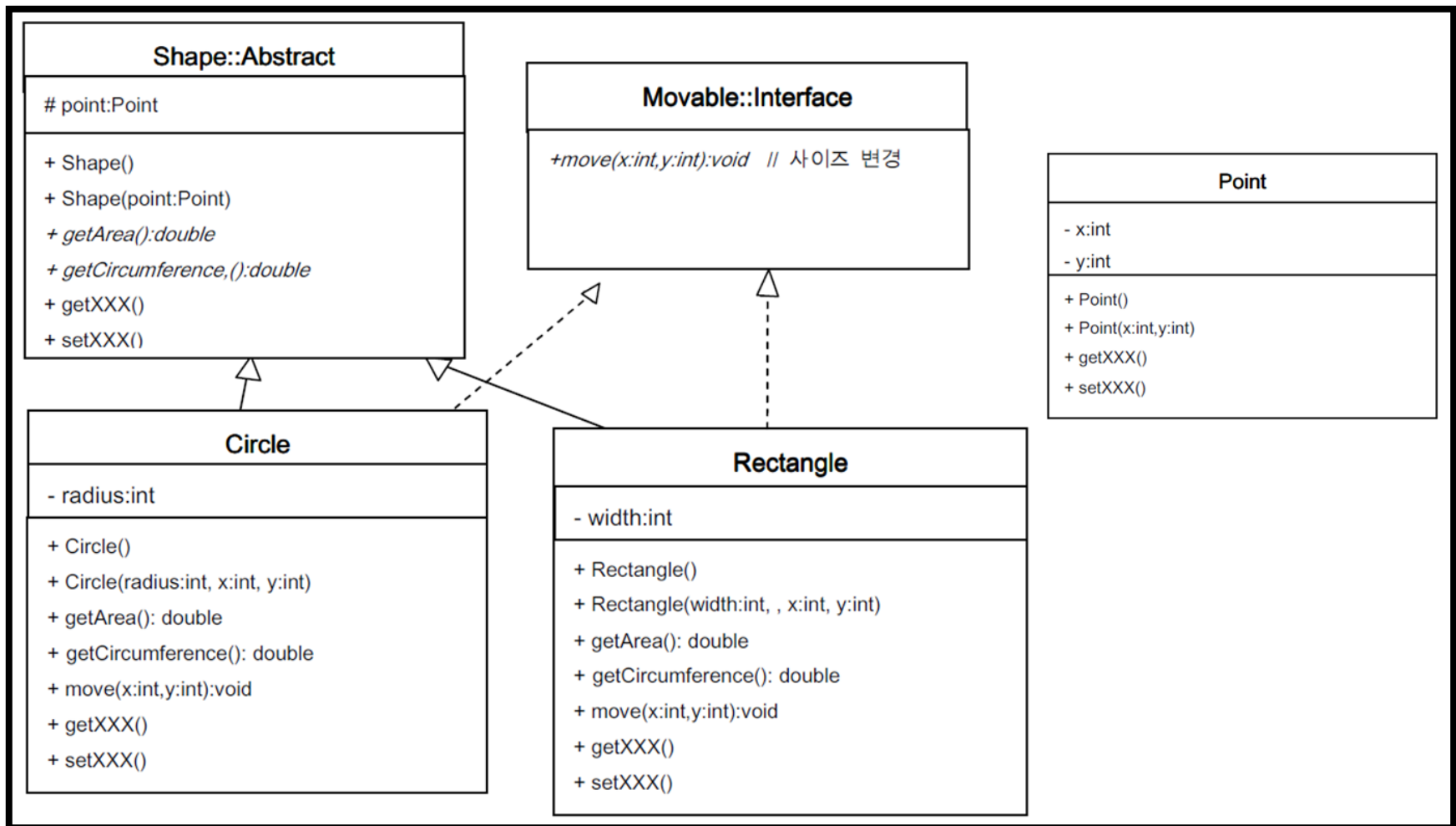
```
public class Company {  
  
    public static void main(String args[]) {  
        HashMap<Integer, Employee> map = new HashMap<Integer, Employee>();  
  
        // 1번의 데이터를 기반으로 객체를 생성하여 HashMap에 넣는다.  
  
        // HashMap에 넣을 때 키 값은 각 객체의 Number로 한다.  
  
        // 모든 객체의 기본 정보를 출력한다(for문을 이용하여 출력한다.)  
  
        // 모든 객체에 인센티브 100씩 지급한다.  
  
        // 모든 객체의 정보와 세금을 출력한다 (for문을 이용하여 출력한다.)  
    }  
}
```

➤ 문제

- 아래의 정보를 가지고 있는 객체를 생성하여 ArrayList에 저장한다.

| 도형 구분 | x | y | width (height) | radius |
|-----------|---|---|----------------|--------|
| Rectangle | 7 | 5 | 4 | |
| Rectangle | 4 | 6 | 5 | |
| Circle | 6 | 7 | | 6 |
| Circle | 8 | 3 | | 7 |

➤ 문제 (cont.)



➤ 문제 (cont.)

| Package명 | Class명 | Method | 설명 |
|----------|-----------|-------------------------------------|---|
| | Shape | +Shape() | 기본 생성자 |
| | | +Shape(point:Point) | 1개의 클래스 변수를 입력 받는 생성자 |
| | | +getArea(): double | 도형의 넓이를 리턴 |
| | | +getCircumference,():double | 도형의 둘레를 리턴 |
| | Movable | +move(x:int,y:int):void | 도형의 좌표를 변경 |
| | Circle | +Circle() | 기본 생성자 |
| | | +Circle(radius:int, x:int, y:int) | 3개의 클래스 변수를 입력 받는 생성자 x, y 좌표를 이용하여 Point객체 생성 |
| | | +getArea(): double | 도형의 넓이를 리턴 |
| | | +getCircumference,():double | 도형의 둘레를 리턴 |
| | | +move(x:int,y:int):void | 도형의 좌표를 변경 |
| | Rectangle | +Rectangle() | 기본 생성자 |
| | | +Rectangle(width:int, x:int, y:int) | 3개의 클래스 변수를 입력 받는 생성자 x, y 좌표를 이용하여 Point객체 생성 |
| | | +getArea(): double | 도형의 넓이를 리턴 |
| | | +getCircumference,():double | 도형의 둘레를 리턴 |
| | | +move(x:int,y:int):void | 도형의 좌표를 변경 |
| | Point | +Point() | 기본 생성자 |
| | | +Point(x:int, y:int) | 2개의 클래스 변수를 입력 받는 생성자 |
| | ShapeTest | + main(args:String[]):void | main 함수 안에서 Shape 객체를 생성 하여 동작 시킨다 |

➤ 문제 (cont.)

Rectangle에서의 move() 함수는 x, y좌표가 입력 한 값보다 2 더 이동한다.

Circle에서의 move() 함수는 x, y좌표가 입력 한 값보다 1 더 이동한다.

```
public class ShapeTest {  
    public static void main(String args[]) {  
  
        // ArrayList를 만들어 [1.사용데이터]의 객체의 순서대로 Circle 객체와 Rectangle  
        객체를 생성하여 넣는다  
        // 모든 객체의 넓이 정보와 둘레 정보를 화면에 출력 한다. - for 문 이용  
        // 모든 객체들에 move() 함수를 이용하여 x가 10증가, y가 10증가  
        되도록 변경 한 후 객체 정보를 화면에 출력 한다. - for 문 이용  
    }  
}
```

➤ 문제 (cont.)

- 넓이와 둘레는 소수점 첫 자리에서 반올림

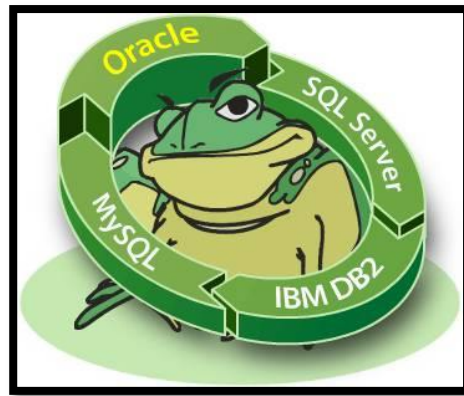
RESULTS

| 구분 | 길이 | x좌표 | y좌표 | Area | Circumference |
|-----------|----|-----|-----|------|---------------|
| Rectangle | 4 | 7 | 5 | 16 | 16 |
| Rectangle | 5 | 4 | 6 | 25 | 20 |
| Circle | 6 | 6 | 7 | 113 | 38 |
| Circle | 7 | 8 | 3 | 154 | 44 |
| 이동 후... | | | | | |
| Rectangle | 4 | 19 | 17 | | |
| Rectangle | 5 | 16 | 18 | | |
| Circle | 6 | 17 | 18 | | |
| Circle | 7 | 19 | 14 | | |

➤ 문제

● 책의 정보를 입력, 수정, 삭제, 검색하는 Program 작성

- 책의 정보는 ISBN, 제목, 작성자, 가격으로 한정
- 사용할 Table 생성 (Table의 이름은 book_tbl)
- Table 생성은 Toad를 이용해서 생성



➤ 문제 (cont.)

● 책의 정보를 입력, 수정, 삭제, 검색하는 Program 작성

- 4권의 책을 Table에 입력 (JDBC Program을 작성해서 입력)
- 단, 4권의 책을 입력하는 작업을 하나의 Transaction으로 처리!!
- 입력된 내용을 console에 출력

| | isbn | title | author | price |
|---|------|-------------|--------|-------|
| ▶ | 1 | 젊은 베르테르의 슬픔 | 괴테 | 5000 |
| | 2 | 파우스트 | 괴테 | 7000 |
| | 3 | 레미제라블 | 빅토르 위고 | 1000 |
| | 4 | 군림천하 | 용대운 | 9000 |

➤ 문제 (cont.)

● 책의 정보를 입력, 수정, 삭제, 검색하는 Program 작성

- 책 제목이 “파우스트” 인 책을 찾아서 가격을 30% 인하.
- 책 저자가 “빅토르위고” 인 책을 찾아서 삭제.
- 위의 두 작업을 하나의 Transaction으로 처리
- 처리된 내용을 console에 출력

| | isbn | title | author | price |
|---|------|-------------|--------|-------|
| ▶ | 1 | 젊은 베르테르의 슬픔 | 괴테 | 5000 |
| | 2 | 파우스트 | 괴테 | 4900 |
| | 4 | 군림천하 | 용대운 | 9000 |

➤ 문제

- 메뉴를 출력한 후 사용자가 원하는 작업 선택 후 처리 결과를 console에 출력한다.

- 메뉴는 다음과 같이 구성하며 종료 메뉴를 선택하기 전까지 반복 수행된다.
- 단, 사용자 입력 값에 대한 validation 처리는 생략한다.
- 기본적인 프로그램의 구성은 MVC Pattern을 근간으로 한다.

```
----- 메뉴를 선택해주세요!! -----  
1. 책 입력  
2. 책 번호로 책 제목 수정  
3. 책 번호로 책 삭제  
4. 책 제목에 대한 키워드 입력 후 찾은 책 출력  
5. 프로그램 종료  
메뉴 번호를 입력하세요.
```


THANKS