프로젝트명 : Homework2_OOP

[문제 1] 다음과 같은 클래스를 작성하시오.

1. 구현 클래스 다이어그램 (Class Diagram)

Shape
- type : int - height : double - width : double - color : String = "white"
+ Shape() + Shape(type:int, height:double, width:double)
+ information() : String + setter() / getter()

ShapeMenu
- sc : Scanner = new Scanner(System.in) - scr : SquareController = new SquareController() - tc : TriangleController = new TriangleController()
+ inputMenu() : void + triangleMenu() :void + squareMenu():void + inputSize(type:int, menuNum:int):void + printInformation(type:int):void

SquareController
- s : Shape = new Shape()
+ calcPerimeter(height:double, width:double) : double + calcArea(height:double, width:double) : double + paintColor(color:String) : void + print() : String

Run
+ main(args:String[]) : void

TriangleController
- s : Shape = new Shape()
+ calcArea(height:double, width:double) : double + paintColor(color:String) : void + print() : String

3. 구현 클래스 설명

Package명	Class명	Method	설명
com.kh.hw. shape.model.vo	Shape	+Shape()	기본 생성자
		+ Shape(type:int, height:double, width:double)	매개변수 있는 생성자
		+getXXX()	저장된 데이터를 불러오는 메소드
		+setXXX()	데이터를 변수에 저장하는 메소드
		+information() : String	높이, 너비, 색깔을 반환하는 메소드
com.kh.hw. shape.controller	SquareController	+calcPerimeter(height:double, width:double) : double	모양 타입 번호와 받은 매개변수를 매개변수 있는 생성자로 초기화 시킨 후 둘레 반환 둘레: 너비*2 + 높이*2
		+calcArea(height:double, width:double) : double	모양 타입 번호와 받은 매개변수를 매개변수 있는 생성자로 초기화 시킨 후 넓이 반환 넓이 : 너비 * 높이
		+paintColor(color:String) : void	setter를 이용해 받아온 매개변수로 값 변경
		+print():String	어떤 모양인지와 Shape의 information()메소드의 반환 값 합쳐 함께 반환
com.kh.hw. shape.controller	TriangleController	+calcArea(height:double, width:double) : double	모양 타입 번호와 받은 매개변수를 매개변수 있는 생성자로 초기화 시킨 후 넓이 반환 넓이 : 너비 * 높이 / 2
		+paintColor(color:String) : void	setter를 이용해 받아온 매개변수로 값 변경
		+print():String	어떤 모양인지와 Shape의 information()메소드의 반환 값 합쳐 함께 반환
com.kh.hw. shape.view	ShapeMenu	+inputMenu():void	삼각형과 사각형을 선택하게 하는 메소드

		+triangleMenu() : void	삼각형 메뉴 출력 메소드
		+squareMenu():void	사각형 메뉴 출력 메소드
		+inputSize(type:int, menuNum:int):void	너비와 높이를 받아 요청사항을 처리하거나 색깔을 받아 요청사항을 처리하는 메소드
		+printInformation(type:int) :void	매개변수에 따라 삼각형/사각형의 정보를 출력하는 메소드
com.kh.hw.shape.run	Run	+main(args:String[]):void	ShapeMenu 객체를 생성 후 inputMenu() 실행

* class 명과 method 명은 변경 하지 않는다.

* 모든 클래스 변수의 getter, setter 함수는 직접 구현한다.

4. class 구조

```
public class Run{

    public static void main(String args[]) {

        // inputMenu() 호출

    }

}
```

```
public class ShapeMenu{
	// 멤버 변수
	public void inputMenu() {
		// ===== 도형 프로그램 =====
		// 3. 삼각형	==> triangleMenu()
		// 4. 사각형	==> squareMenu()
		// 9. 프로그램 종료 => "프로그램 종료" 출력 후 프로그램 종료
		// 메뉴 번호 :
		// 잘못 입력했을 시 "잘못된 번호입니다. 다시 입력해주세요." 출력 후 반복
	}
	public void triangleMenu(){
		// ===== 삼각형 =====
		// 1. 삼각형 면적 ==> inputSize()
		// 2. 삼각형 색칠 ==> inputSize()
		// 3. 삼각형 정보 ==> printInformation()
		// 9. 메인으로 ==> "메인으로 돌아갑니다." 출력 후 inputMenu()로
		// 메뉴 번호 :
		// 잘못 입력했을 시 "잘못된 번호입니다. 다시 입력해주세요." 출력 후 반복
	}
	public void squareMenu(){
		// ===== 사각형 =====
		// 1. 사각형 둘레 ==> inputSize()
		// 2. 사각형 면적 ==> inputSize()
		// 3. 사각형 색칠 ==> inputSize()
		// 4. 사각형 정보 ==> printInformation()
		// 9. 메인으로 ==> "메인으로 돌아갑니다." 출력 후 inputMenu()로
		// 메뉴 번호 :
		// 잘못 입력했을 시 "잘못된 번호입니다. 다시 입력해주세요." 출력 후 반복
	}
	// 삼각형 메뉴, 사각형 메뉴의 세부 메뉴에서 모두 같은 메소드로 이동하기 때문에
	// 삼각형인지 사각형인지, 몇 번 메뉴인지 구분하기 위해 매개변수로 넘겨줌
```

```
	public void inputSize(int type, int menuNum) {
		// 매개변수로 들어온 type과 menuNum의 숫자에 따라 출력이 달라짐

		// int type이 '삼각형'이면서 menuNum이 1번일 경우
		// 높이 :
		// 너비 :
		// 삼각형 면적 : ==> tc(TriangleController)의 calcArea() 출력
		// int type이 '삼각형'이면서 menuNum이 2번일 경우
		// 색깔을 입력하세요 :
		// tc의 paintColor() 호출 후 "색이 수정되었습니다" 출력

		// int type이 '사각형'이면서 menuNum이 1번이나 2번일 경우
		// 높이 :
		// 너비 :
		// menuNum이 1번일 경우
		// 사각형 둘레 : ==> scr(SquareController)의 calcPerimeter() 출력
		// menuNum이 2번일 경우
		// 사각형 면적 : ==> scr의 calcArea() 출력
		// int type이 '사각형'이면서 menuNum이 3번일 경우
		// 색깔을 입력하세요 :
		// scr의 paintColor() 호출 후 "색이 수정되었습니다" 출력
	}
	public void printInformation(int type){
		// int type에 따라 print()메소드를 불러오는 controller가 다름
		// int type이 '삼각형'일 경우 tc.print() 출력
		// int type이 '사각형'일 경우 scr.print() 출력
	}
}
```

```
public class TriangleController{
	// 멤버 변수
	public double calcArea(double height, double width) {
		// 매개변수로 넘어온 값을 Shape의 매개변수 있는 생성자에 넣어
		// Shape의 필드들 초기화하고 면적 계산법을 통해 계산된 값 반환
	}
	public void paintColor(String color){
		// setter를 통해 매개변수로 넘어온 값으로 변경
	}
	public String print(){
		// "삼각형" + s.information()으로 삼각형의 정보 리턴
	}
}
public class SquareController{
	// 멤버 변수
	public double calcPerimeter(double height, double width) {
		// 매개변수로 넘어온 값을 Shape의 매개변수 있는 생성자에 넣어
		// Shape의 필드들 초기화
		// 둘레 계산법을 통해 계산된 값 반환
	}
	public double calcArea(double height, double width) {
		// 매개변수로 넘어온 값을 Shape의 매개변수 있는 생성자에 넣어
		// Shape의 필드들 초기화
		// 면적 계산법을 통해 계산된 값 반환
	}
	public void paintColor(String color){
		// setter를 통해 매개변수로 넘어온 값으로 변경
	}
	public String print(){
		// "사각형" + s.information()으로 삼각형의 정보 리턴
	}
}
```

5. 실행 결과

```
===== 도형 프로그램 =====
3. 삼각형
4. 사각형
9. 프로그램 종료
메뉴 번호 : 1
잘못된 번호입니다. 다시 입력하세요.
===== 도형 프로그램 =====
3. 삼각형
4. 사각형
9. 프로그램 종료
메뉴 번호 : 3
===== 삼각형 =====
1. 삼각형 면적
2. 삼각형 색칠
3. 삼각형 정보
9. 메인으로
메뉴 번호 : 1
높이 : 10
너비 : 3
삼각형 면적 : 15.0
===== 삼각형 =====
1. 삼각형 면적
2. 삼각형 색칠
3. 삼각형 정보
9. 메인으로
메뉴 번호 : 3
삼각형 10.0 3.0 white
===== 삼각형 =====
1. 삼각형 면적
2. 삼각형 색칠
3. 삼각형 정보
9. 메인으로
메뉴 번호 : 2
색깔을 입력하세요 : red
```

```
색이 수정되었습니다.
===== 삼각형 =====
1. 삼각형 면적
2. 삼각형 색칠
3. 삼각형 정보
9. 메인으로
메뉴 번호 : 3
삼각형 10.0 3.0 red
===== 삼각형 =====
1. 삼각형 면적
2. 삼각형 색칠
3. 삼각형 정보
9. 메인으로
메뉴 번호 : 9
메인으로 돌아갑니다.
===== 도형 프로그램 =====
3. 삼각형
4. 사각형
9. 프로그램 종료
메뉴 번호 : 4
===== 사각형 =====
1. 사각형 둘레
2. 사각형 면적
3. 사각형 색칠
4. 사각형 정보
9. 메인으로
메뉴 번호 : 1
높이 : 10
너비 : 3
사각형 둘레 : 26.0
===== 사각형 =====
1. 사각형 둘레
2. 사각형 면적
3. 사각형 색칠
4. 사각형 정보
9. 메인으로
메뉴 번호 : 2
높이 : 10
너비 : 3
사각형 둘레 : 26.0
===== 사각형 =====
1. 사각형 둘레
2. 사각형 면적
3. 사각형 색칠
4. 사각형 정보
9. 메인으로
메뉴 번호 : 2
```

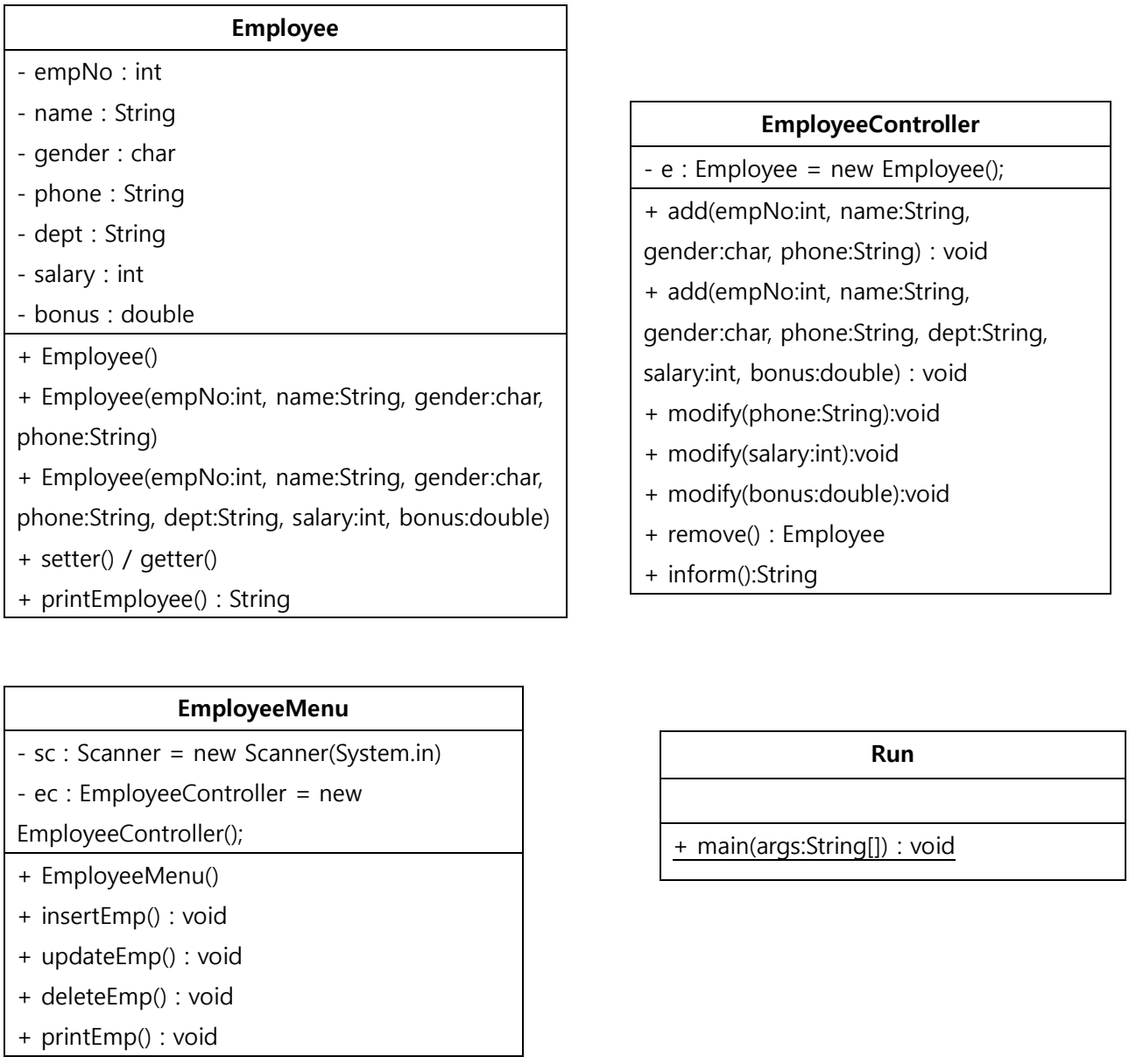
```
높이 : 10
너비 : 3
사각형 면적 : 30.0
===== 사각형 =====
1. 사각형 둘레
2. 사각형 면적
3. 사각형 색칠
4. 사각형 정보
9. 메인으로
메뉴 번호 : 4
사각형 10.0 3.0 white
===== 사각형 =====
1. 사각형 둘레
2. 사각형 면적
3. 사각형 색칠
4. 사각형 정보
9. 메인으로
메뉴 번호 : 3
색깔을 입력하세요 : blue
색이 수정되었습니다.
===== 사각형 =====
1. 사각형 둘레
2. 사각형 면적
3. 사각형 색칠
4. 사각형 정보
9. 메인으로
메뉴 번호 : 4
사각형 10.0 3.0 blue
===== 사각형 =====
1. 사각형 둘레
2. 사각형 면적
3. 사각형 색칠
4. 사각형 정보
9. 메인으로
메뉴 번호 : 9
메인으로 돌아갑니다.
===== 도형 프로그램 =====
3. 삼각형
4. 사각형
9. 프로그램 종료
메뉴 번호 : 9
프로그램을 종료합니다.
```

[문제 2] 다음과 같은 조건을 만족하는 프로그램을 작성 하시오

1. 사용 데이터

empNo	name	gender	phone	dept	salary	bonus
1	홍길동	남	010-1234-5678			
2	김길순	여	010-5678-1234	영업부	3000000	0.15

2. 구현 클래스 다이어그램 (Class Diagram)



3. 구현 클래스 설명

Package명	Class명	Method	설명
com.kh.hw. employee.model.vo	Employee	+Employee()	기본 생성자
		+Employee(empNo:int, name:String, gender:char, phone:String)	4개의 초기 값을 받는 생성자
		+Employee(empNo:int, name:String, gender:char, phone:String, dept:String, salary:int, bonus:double)	7개의 초기 값을 받는 생성자
		+printEmployee():String	직원 정보 반환
com.kh.hw. employee.controller	EmployeeController	+ add(empNo:int, name:String, gender:char, phone:String) : void	매개변수 있는 생성자를 이용하여 데이터 저장하는 메소드
		+ add(empNo:int, name:String, gender:char, phone:String, dept:String, salary:int, bonus:double) : void	매개변수 있는 생성자를 이용하여 데이터 저장하는 메소드
		+modify(phone:String):void	setter로 정보 수정
		+ modify(salary:int):void	setter로 정보 수정
		+ modify(bonus:double) :void	setter를 이용하여 정보 수정
		+ remove() : Employee	객체를 삭제하는 메소드
		+ inform():String	객체에 저장된 데이터를 가져와 반환하는 메소드
com.kh.hw. employee.view	EmployeeMenu	+ EmployeeMenu()	메인 메뉴를 출력하는 기본 생성자
		+ insertEmp() : void	저장할 데이터를 사용자에게 받는 메소드
		+ updateEmp():void	수정할 데이터를 사용자에게 받는 메소드
		+ deleteEmp() : void	데이터 삭제하는 메소드
		+ printEmp():void	데이터 출력하는 메소드
com.kh.hw. employee.run	Run	+ main(args:String[]):void	EmployeeMenu 실행

* class 명과 method 명은 변경 하지 않는다.

* 모든 클래스 변수의 getter, setter 함수는 직접 구현한다.

4. class 구조

```
public class Run{

        public static void main(String args[]) {

                EmployeeMenu em = new EmployeeMenu();

    }

}
```

```
public class EmployeeMenu{

        // 멤버 필드

        public EmployeeMenu() {

                // 1. 사원 추가 ==> insertEmp()

                // 2. 사원 수정 ==> updateEmp()

                // 3. 사원 삭제 ==> deleteEmp()

                // 4. 사원 출력 ==> printEmp()

                // 9. 프로그램 종료 ==> "프로그램을 종료합니다." 출력 후 프로그램 종료

                // 메뉴 번호를 누르세요 :

                // 번호를 잘못 입력했으면 잘못 입력했다는 안내가 뜸

        }
```

```
public void insertEmp(){
        // 사원 번호 :
        // 사원 이름 :
        // 사원 성별 :
        // 전화 번호 :
        // 추가 정보를 더 입력하시겠습니까?(y/n) :
        // 추가적인 정보를 더 입력한다고 했을 시(y 또는 Y)
        // 사원부서, 사원 연봉, 보너스 율을 추가로 더 받고
        // 모든 데이터를 EmployeeController의 add메소드 인자로 보냄
        // 추가정보를 입력하지 않겠다고 하면 기본정보만 add메소드 인자로 보냄
}
public void updateEmp(){
        // 가장 최신의 사원 정보를 수정하게 됩니다.
        // 사원의 어떤 정보를 수정하시겠습니까?
        // 1. 전화번호
        // 2. 사원 연봉
        // 3. 보너스 율
        // 9. 돌아가기
        // 메뉴 번호를 누르세요 :
        // 사용자가 수정하고 싶은 내용에 대한 번호를 입력하면
        // 수정할 XXX : 라고 안내문을 출력 후 사용자에게 값을 받고
        // 받은 값을 EmployeeController의 modify() 인자로 넣어 전달
        // 9번을 입력하면 "메인메뉴로 돌아갑니다" 출력 후 메인 메뉴로
        // 잘못 입력할 경우 "잘못 입력하셨습니다." 출력 후 메인메뉴로
}
```

```
        public void deleteEmp(){
                // 정말 삭제할 것인지 물어본 후 삭제하겠다고 하면
                // EmployeeController의 remove()메소드를 호출하여 반환 값에 따라
                // 데이터 삭제에 실패하였는지 성공하였는지 출력
        }
        public void printEmp(){
                // 사원정보가 있다면 사원정보 출력, 없다면 "사원 데이터가 없습니다"출력
        }
}
```

```
public class EmployeeController{
        // 멤버 필드
        public void add(/* 매개변수 생략 */) {
                // 받아온 매개변수만큼 매개변수 있는 생성자를 통해 값 저장
        }
        public void modify(/* 매개변수 생략 */){
                // 받아온 매개변수를 이용하여 해당 정보 수정
        }
        public Employee remove(){
                // 객체 e에 null을 저장하여 객체 삭제
        }
        public String inform(){
                // 객체 e가 null이라면 null 반환, 아니라면 사원 정보 반환
        }
}
```

5. 실행 결과

```
1. 사원 추가
2. 사원 수정
3. 사원 삭제
4. 사원 출력
9. 프로그램 종료
메뉴 번호를 누르세요 : 1
사원 번호 : 1
사원 이름 : 홍길동
사원 성별 : 남
전화 번호 : 010-1234-5678
추가 정보를 더 입력하시겠습니까?(y/n) : n
1. 사원 추가
2. 사원 수정
3. 사원 삭제
4. 사원 출력
9. 프로그램 종료
메뉴 번호를 누르세요 : 4
1 홍길동 남 010-1234-5678 null 0 0.0
1. 사원 추가
2. 사원 수정
3. 사원 삭제
4. 사원 출력
9. 프로그램 종료
메뉴 번호를 누르세요 : 1
사원 번호 : 2
사원 이름 : 김길순
사원 성별 : 여
전화 번호 : 010-5678-1234
추가 정보를 더 입력하시겠습니까?(y/n) : y
사원 부서 : 영업부
사원 연봉 : 3000000
보너스 율 : 0.5
1. 사원 추가
2. 사원 수정
3. 사원 삭제
4. 사원 출력
9. 프로그램 종료
메뉴 번호를 누르세요 : 4
2 김길순 여 010-5678-1234 영업부 3000000 0.5
1. 사원 추가
2. 사원 수정
3. 사원 삭제
4. 사원 출력
9. 프로그램 종료
메뉴 번호를 누르세요 : 2
```

```
가장 최신의 사원 정보를 수정하게 됩니다.
사원의 어떤 정보를 수정하시겠습니까?
1. 전화 번호
2. 사원 연봉
3. 보너스 율
9. 돌아가기
메뉴 번호를 누르세요 : 3
수정할 보너스율 :0.15
1. 사원 추가
2. 사원 수정
3. 사원 삭제
4. 사원 출력
9. 프로그램 종료
메뉴 번호를 누르세요 : 4
2 김길순 여 010-5678-1234 영업부 3000000 0.15
1. 사원 추가
2. 사원 수정
3. 사원 삭제
4. 사원 출력
9. 프로그램 종료
메뉴 번호를 누르세요 : 2
가장 최신의 사원 정보를 수정하게 됩니다.
사원의 어떤 정보를 수정하시겠습니까?
1. 전화 번호
2. 사원 연봉
3. 보너스 율
9. 돌아가기
메뉴 번호를 누르세요 : 9
메인메뉴로 돌아갑니다.
1. 사원 추가
2. 사원 수정
3. 사원 삭제
4. 사원 출력
9. 프로그램 종료
메뉴 번호를 누르세요 : 3
정말 삭제하시겠습니까? (y/n) : y
데이터 삭제에 성공하였습니다.
1. 사원 추가
2. 사원 수정
3. 사원 삭제
4. 사원 출력
9. 프로그램 종료
메뉴 번호를 누르세요 : 4
```

```
사원 데이터가 없습니다.
1. 사원 추가
2. 사원 수정
3. 사원 삭제
4. 사원 출력
9. 프로그램 종료
메뉴 번호를 누르세요 : 9
프로그램을 종료합니다.
```