

프로젝트명 : Homework2_OOP

[문제 1] 다음과 같은 클래스를 작성하시오.

1. 구현 클래스 다이어그램 (Class Diagram)

Shape	ShapeMenu
- type : int - height : double - width : double - color : String = "white"	- sc : Scanner = new Scanner(System.in) - scr : SquareController = new SquareController() - tc : TriangleController = new TriangleController()
+ Shape() + Shape(type:int, height:double, width:double) + information() : String + setter() / getter()	+ inputMenu() : void + triangleMenu() : void + squareMenu() : void + inputSize(type:int, menuNum:int) : void + printInformation(type:int) : void
SquareController	Run
- s : Shape = new Shape() + calcPerimeter(height:double, width:double) : double + calcArea(height:double, width:double) : double + paintColor(color:String) : void + print() : String	+ 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 Shape 의 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 Shape 의 information()메소드의 반환 값 합쳐 함께 반환
com.kh.hw. shape.view	ShapeMenu	+inputMenu():void	삼각형과 사각형을 선택 하게 하는 메소드

		+triangleMenu() : void	삼각형 메뉴 출력 메소드
		+squareMenu():void	사각형 메뉴 출력 메소드
		+inputSize(type:int, menuNum:int):void	너비와 높이를 받아 요청 사항을 처리하거나 색깔 을 받아 요청사항을 처리 하는 메소드
		+printInformation(type:int) :void	매개변수에 따라 삼각형/ 사각형의 정보를 출력하 는 메소드
com.kh.hw. shape.run	Run	<u>+main(args:String[]):void</u>	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 사각형의 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
메인으로 돌아갑니다.
===== 도형 프로그램 =====
3. 삼각형
4. 사각형
9. 프로그램 종료
메뉴 번호 : 9
프로그램을 종료합니다.
```