

# REPORT 2



## 소프트웨어프로젝트 04 분반

담당 교수: 이남규 교수님  
학 과: 영어영문학과  
학 번: 20151816  
이 름: 차유진  
제 출 일: 2020.04.21

# 1. 사각형의 넓이와 둘레

## A. 문제

사각형을 나타내는 Rectangle 클래스를 작성하라.

- 사각형의 가로와 세로를 위한 double 자료형 변수 width, height, 각 디폴트 값은 1 이다.
- 사각형의 색깔을 위한 문자열 데이터 필드 color, 디폴트 값은 “white”
- 디폴트 값의 사각형을 위한 무인자 생성자
- 주어진 가로와 세로의 사각형을 위한 생성자
- 주어진 하나의 길이(가로와 세로로 사용)의 사각형을 위한 생성자
- 사각형의 면적을 계산하는 getArea() 메소드
- 사각형의 둘레의 길이를 계산하는 getPerimeter() 메소드

- ① Rectangle 객체 배열(크기 3)을 선언하고, 세가지 다른 생성자를 호출하여 객체가 생성되도록 한다.
- ② 생성자 함수에서 다른 생성자 호출이 되도록 this()를 이용한다.
- ③ getArea() 메소드, getPerimeter() 메소드를 호출하여 사각형의 면적과 둘레를 출력한다.

## B. 해결 방법

1. 매개변수에 따른 생성자를 각각 구성한다.
2. 넓이와 둘레를 구하는 메소드를 생성한다.
3. 메인메소드에서, 크기가 3 인 객체배열을 선언하고 세가지 다른 생성자를 호출하도록 한다.
4. 각 사각형의 넓이와 둘레를 출력한다.

## C. 소스코드 및 주석

```
/*
파일명: Rectangle.java
입력: -
출력: 배열에 저장된 사각형 1,2,3 각각에 대한 넓이와 둘레

*/
public class Rectangle {
    double width;
    double height;
    String color;

    public Rectangle() { // 아무런 값이 없는 경우
        this(1,1,"white"); // 디폴트 값 지정
    }

    public Rectangle(double width, String color) { // 숫자 하나, 색상만 입력 받은 경우
        this(width, width, color); // 가로와 세로가 같은 정사각형
    }

    public Rectangle (double width, double height, String color) { // 가로, 세로, 색깔 각각 지정받은 경우
        this.width = width;
        this.height = height;
        this.color = color;
    }
}
```

```

public double getArea(){ //넓이 = 가로 * 세로
    return height * width;
}

public double getPerimeter(){ //둘레 = 가로 * 2 + 세로 * 2
    return width*2 + height*2;
}

public static void main(String[] args) { //메인 메소드

    Rectangle [] rectangle = new Rectangle[3]; //공간이 세 개인 객체 배열 생성

    rectangle[0] = new Rectangle(3, 5, "black"); //사각형 1: 가로 3, 세로 5, 색깔 black
    rectangle[1] = new Rectangle(4, "red"); //사각형 2: 가로 4, (세로 생략 => 가로와 동일하게 4), 색깔 red
    rectangle[2] = new Rectangle(); //값을 따로 지정하지 않음 => 디폴트 값 (가로 1, 세로 1, 색깔 white)

    for(int i=0; i<3; i++) { //사각형 세 개 출력하기 위해 세 번 반복
        double area = rectangle[i].getArea(); //i번째 사각형 객체에 대한 면적 가져오기
        System.out.println((i+1)+"번 사각형("+rectangle[i].width+", "+rectangle[i].height+", "+rectangle[i].color+")의 면적:
"+area);

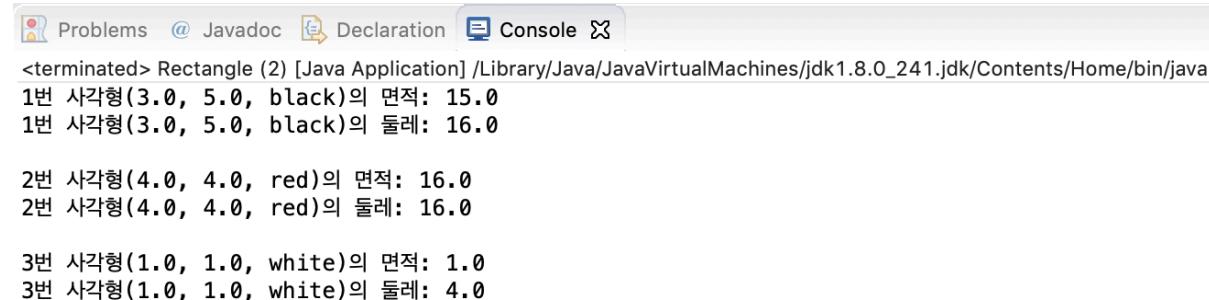
        double perimeter = rectangle[i].getPerimeter(); //i번째 사각형 객체에 대한 둘레 가져오기
        System.out.println((i+1)+"번 사각형("+rectangle[i].width+", "+rectangle[i].height+", "+rectangle[i].color+")의 둘레:
"+perimeter);

        System.out.println();
    }

}

```

## D. 입력 및 출력 결과 예시



The screenshot shows the Java application running in an IDE. The console tab displays the following output:

```

Problems @ Javadoc Declaration Console
<terminated> Rectangle (2) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_241.jdk/Contents/Home/bin/java
1번 사각형(3.0, 5.0, black)의 면적: 15.0
1번 사각형(3.0, 5.0, black)의 둘레: 16.0

2번 사각형(4.0, 4.0, red)의 면적: 16.0
2번 사각형(4.0, 4.0, red)의 둘레: 16.0

3번 사각형(1.0, 1.0, white)의 면적: 1.0
3번 사각형(1.0, 1.0, white)의 둘레: 4.0

```

## 2-1. 계산기 1(한 개의 패키지)

### A. 문제

다수의 클래스를 만들고 활용하는 연습을 해보자. 더하기(+), 빼기(-), 곱하기(\*) 나누기(/)를 수행하는 각 클래스 Add, Sub, Mul, Div 를 만들어라. 이들은 모두 다음과 같은 필드와 메소드들을 가진다.

- int 타입의 a, b 필드 : 2 개의 피연산자
- void setValue(int a, int b) : 피연산자 값을 객체 내에 저장
- int calculate() : 클래스의 목적에 맞는 연산을 실행하고 결과를 리턴한다.

Calc 클래스에 있는 main() 메소드에서는 다음 실행 사례와 같이 두 정수와 연산자를 입력받고 Add, Sub, Mul, Div 중에서 이 연산을 실행할 수 있는 객체를 생성하고 setValue()와 calculate()를 호출하여 출력하도록 작성하라. For-loop 를 이용하여 6 번 반복 연산을 수행하도록 하여 아래 실행 결과를 모두 보여줄 수 있도록 하라.

\* default 패키지에 하나 파일에는 Calc, 다른 하나의 파일에는 Add, Sub, Mul, Div 클래스를 포함하도록 작성하라.

### B. 해결 방법

1. 디폴트 패키지에 두개의 파일- 메인메소드가 들어갈 Calc.java 와 연산 함수가 들어갈 Operators.java
  - 생성한다.
2. Operators.java 파일에 각각 연산자를 위한 Add, Sub, Mul, Div 클래스를 생성한다.
3. For 문을 이용하여 두 정수와 연산자를 총 6 번 입력 받도록 한다.
4. if 문을 이용하여 입력 받은 연산자에 따라 각 연산자 클래스를 호출한다.
5. 나눗셈을 0 으로 할 경우, 연산자가 +, -, \*, / 외의 것일 경우 예외처리를 하여 오류 메세지를 띄운다.

### C. 소스코드 및 주석

```
/*
파일명: Calc.java
입력: 두 정수와 연산자
출력: 연산자에 따른 두 정수의 연산 결과 6회

*/
import java.util.Scanner;

public class Calc {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        for(int i=0;i<6;i++) { // 6번 반복
            System.out.print("두 정수의 연산식을 입력하시오(5 + 7) >> ");
            int a = scanner.nextInt(); // 첫 번째 정수 입력
            char cal = scanner.next().charAt(0); // 연산자 입력
            int b = scanner.nextInt(); // 두 번째 정수 입력

            if(cal=='+') { // 입력받은 연산자가 더하기 일 경우
                Add add = new Add(); // Add 클래스 호출
                System.out.println(add.calculate(a, b));
            }
            else if(cal=='-') { // 입력받은 연산자가 빼기 일 경우
                Sub sub = new Sub(); // Sub 클래스 호출
                System.out.println(sub.calculate(a, b));
            }
            else if(cal=='*') { // 입력받은 연산자가 곱하기 일 경우
                Mul mul = new Mul(); // Mul 클래스 호출
                System.out.println(mul.calculate(a, b));
            }
            else if(cal=='/') { // 입력받은 연산자가 나누기 일 경우
                Div div = new Div(); // Div 클래스 호출
                System.out.println(div.calculate(a, b));
            }
            else {
                System.out.println("잘못된 연산자입니다.");
            }
        }
    }
}
```

```

        add.setValue(a, b);
        System.out.println(add.calculate());
    }

    else if(cal=='-') { // 입력받은 연산자가 빼기 일 경우
        Sub sub = new Sub(); // Sub 클래스 호출
        sub.setValue(a, b);
        System.out.println(sub.calculate());
    }

    else if(cal=='*') { // 입력받은 연산자가 곱하기 일 경우
        Mul mul = new Mul(); // Mul 클래스 호출
        mul.setValue(a, b);
        System.out.println(mul.calculate());
    }

    else if(cal=='/') { // 입력받은 연산자가 나누기 일 경우
        if(b==0) System.out.println("0으로 나눌 수 없습니다."); // 두 번째 입력받은 정수가 0 이면 오류 메세지 표시
        Div div = new Div(); // Div 클래스 호출
        div.setValue(a, b);
        System.out.println(div.calculate());
    }

    else System.out.println("잘못된 연산자입니다."); // 연산자가 +-*/ 중 하나가 아닐 경우 오류 메세지 표시
}
scanner.close();
}

/*
파일명: Operators.java
입력: -
출력: Calc.java 파일에 각 연산자 클래스의 결과값 리턴
*/

```

```

class Add {
    int a,b;
    void setValue(int a, int b){
        this.a = a;
        this.b = b;
    }
    int calculate(){
        return a+b;
    }
}

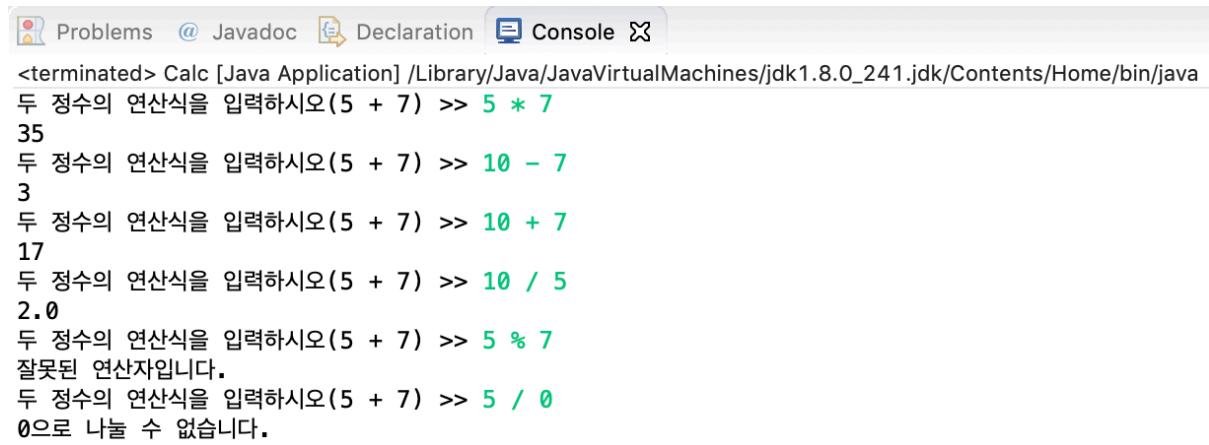
class Sub {
    int a,b;
    void setValue(int a, int b){
        this.a = a;
        this.b = b;
    }
    int calculate(){
        return a-b;
    }
}

class Mul {
    int a,b;
    void setValue(int a, int b){
        this.a = a;
        this.b = b;
    }
    int calculate(){
        return a*b;
    }
}

```

```
class Div {  
    int a,b;  
    void setValue(int a, int b){  
        this.a = a;  
        this.b = b;  
    }  
    double calculate(){  
        return (double)a/b;  
    }  
}
```

## D. 입력 및 출력 결과 예시



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The console window displays the following interaction:

```
<terminated> Calc [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_241.jdk/Contents/Home/bin/java  
두 정수의 연산식을 입력하시오(5 + 7) >> 5 * 7  
35  
두 정수의 연산식을 입력하시오(5 + 7) >> 10 - 7  
3  
두 정수의 연산식을 입력하시오(5 + 7) >> 10 + 7  
17  
두 정수의 연산식을 입력하시오(5 + 7) >> 10 / 5  
2.0  
두 정수의 연산식을 입력하시오(5 + 7) >> 5 % 7  
잘못된 연산자입니다.  
두 정수의 연산식을 입력하시오(5 + 7) >> 5 / 0  
0으로 나눌 수 없습니다.
```

## 2-2. 계산기 2(두 개의 패키지)

### A. 문제

다수의 클래스를 만들고 활용하는 연습을 해보자. 더하기(+), 빼기(-), 곱하기(\*) 나누기(/)를 수행하는 각 클래스 Add, Sub, Mul, Div 를 만들어라. 이들은 모두 다음과 같은 필드와 메소드들을 가진다.

- int 타입의 a, b 필드 : 2 개의 피연산자
- void setValue(int a, int b) : 피연산자 값을 객체 내에 저장
- int calculate() : 클래스의 목적에 맞는 연산을 실행하고 결과를 리턴한다.

Calc 클래스에 있는 main() 메소드에서는 다음 실행 사례와 같이 두 정수와 연산자를 입력받고 Add, Sub, Mul, Div 중에서 이 연산을 실행할 수 있는 객체를 생성하고 setValue()와 calculate()를 호출하여 출력하도록 작성하라. For-loop 를 이용하여 6 번 반복 연산을 수행하도록 하여 아래 실행 결과를 모두 보여줄 수 있도록 하라.

\* pkg1 패키지에 Calc 클래스, pkg2 패키지에 Add, Sub, Mul, Div 클래스를 가지도록 작성하라

### B. 해결 방법

1. 두개의 패키지 pkg1 과 pkg2 를 만들고, pkg1 에는 Calc2.java 파일, pkg2 에는 Add.java, Sub.java, Mul.java, Div.java 파일을 만든다.
2. Calc2.java 파일에서 pkg2 의 파일들을 import 해준다.
3. pkg2 에서 각 파일에 해당 연산자의 함수를 넣어준다.
4. Calc2 에서 For 문을 이용하여 두 정수와 연산자를 총 6 번 입력 받도록 한다.
5. if 문을 이용하여 입력 받은 연산자에 따라 각 연산자 클래스를 호출한다.
6. 나눗셈을 0 으로 할 경우, 연산자가 +, -, \*, / 외의 것일 경우 예외처리를 하여 오류 메세지를 띄운다.

### C. 소스코드 및 주석

```
/*
파일명: Calc2.java
입력: 두 정수와 연산자
출력: 연산자에 따른 두 정수의 연산 결과 6회

*/
package pkg1;

import java.util.Scanner;
import pkg2.Add; // pkg2에 있는 연산자 파일들 불러오기
import pkg2.Sub;
import pkg2.Mul;
import pkg2.Div;

public class Calc2 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        for(int i=0;i<6;i++) { // 6번 반복
```

```

System.out.print("두 정수의 연산식을 입력하시오(5 + 7) >> ");
int a = scanner.nextInt(); // 첫 번째 정수 입력
char cal = scanner.next().charAt(0); // 연산자 입력
int b = scanner.nextInt(); // 두 번째 정수 입력

if(cal=='+') { // 입력받은 연산자가 더하기 일 경우
    Add add = new Add(); // Add 클래스 호출
    add.setValue(a, b);
    System.out.println(add.calculate());
}
else if(cal=='-') { // 입력받은 연산자가 빼기 일 경우
    Sub sub = new Sub(); // Sub 클래스 호출
    sub.setValue(a, b);
    System.out.println(sub.calculate());
}
else if(cal=='*') { // 입력받은 연산자가 곱하기 일 경우
    Mul mul = new Mul(); // Mul 클래스 호출
    mul.setValue(a, b);
    System.out.println(mul.calculate());
}
else if(cal=='/') { // 입력받은 연산자가 나누기 일 경우
    if(b==0) System.out.println("0으로 나눌 수 없습니다."); // 두 번째 입력받은 정수가 0 이면 오류 메세지 표시
    Div div = new Div(); // Div 클래스 호출
    div.setValue(a, b);
    System.out.println(div.calculate());
}
else System.out.println("잘못된 연산자입니다."); // 연산자가 +-*/ 중 하나가 아닐 경우 오류 메세지 표시
}
scanner.close();
}

/*
파일명: Add.java
입력: -
출력: Calc2.java 파일에 나누기 연산자에 대한 결과값 리턴
*/
package pkg2;

public class Add {
    int a,b;
    public void setValue(int a, int b){
        this.a = a;
        this.b = b;
    }
    public int calculate(){
        return a+b;
    }
}

/*
파일명: Sub.java
입력: -
출력: Calc2.java 파일에 빼기 연산자에 대한 결과값 리턴
*/
package pkg2;

public class Sub {
    int a,b;
    public void setValue(int a, int b){
        this.a = a;
        this.b = b;
    }
    public int calculate(){
        return a-b;
    }
}

```

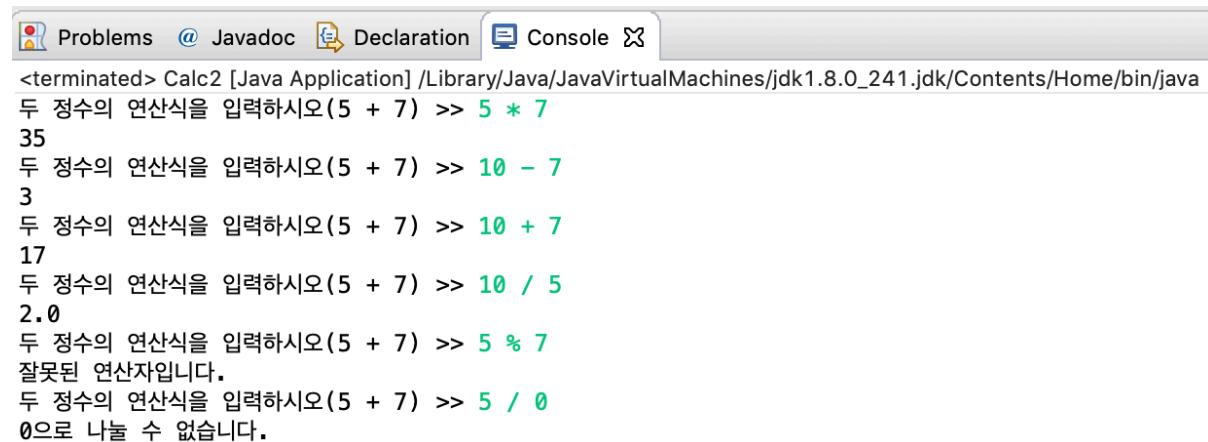
```
/*
파일명: Mul.java
입력: -
출력: Calc2.java 파일에 곱하기 연산자에 대한 결과값 리턴
*/
package pkg2;

public class Mul {
    int a,b;
    public void setValue(int a, int b){
        this.a = a;
        this.b = b;
    }
    public int calculate(){
        return a*b;
    }
}

/*
파일명: Div.java
입력: -
출력: Calc2.java 파일에 더하기 연산자에 대한 결과값 리턴
*/
package pkg2;

public class Div {
    int a,b;
    public void setValue(int a, int b){
        this.a = a;
        this.b = b;
    }
    public double calculate(){
        return (double)a/b;
    }
}
```

## D. 입력 및 출력 결과 예시



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following interaction:

```
<terminated> Calc2 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_241.jdk/Contents/Home/bin/java
두 정수의 연산식을 입력하시오(5 + 7) >> 5 * 7
35
두 정수의 연산식을 입력하시오(5 + 7) >> 10 - 7
3
두 정수의 연산식을 입력하시오(5 + 7) >> 10 + 7
17
두 정수의 연산식을 입력하시오(5 + 7) >> 10 / 5
2.0
두 정수의 연산식을 입력하시오(5 + 7) >> 5 % 7
잘못된 연산자입니다.
두 정수의 연산식을 입력하시오(5 + 7) >> 5 / 0
0으로 나눌 수 없습니다.
```

### 3. 추상클래스의 활용

#### A. 문제

다음은 키(key)와 값(value)을 하나의 아이템으로 저장하고 검색 수정이 가능한 추상 클래스가 있다.

```
abstract class PairMap {
    protected String keyArray []; // key 들을 저장하는 배열
    protected String valueArray []; // value 들을 저장하는 배열
    abstract String get(String key); // key 값으로 value 를 검색
    abstract void put(String key, String value); // key 와 value 를 쌍으로 저장
    abstract String delete(String key); // key 값을 가진 아이템(value 와 함께)을 삭제. 삭제된 value 값 리턴
    abstract int length(); // 현재 저장된 아이템의 개수 리턴
}
```

PairMap 을 상속받는 Dictionary 클래스를 구현하고, 이를 다음과 같이 활용하는 main() 메소드를 가진 DictionaryApp 도 작성하라.

```
public static void main(String[] args) {
    Dictionary dic = new Dictionary(10);
    dic.put("황기태", "자바");
    dic.put("이재문", "파이썬");
    dic.put("이재문", "C++"); // 이재문의 값을 C++로 수정
    System.out.println("이재문의 값은 " + dic.get("이재문"));
    System.out.println("황기태의 값은 " + dic.get("황기태"));
    dic.delete("황기태");
    System.out.println("황기태의 값은 " + dic.get("황기태"));
}
```

#### B. 해결 방법

1. 추상클래스 PairMap 과 메인 메소드를 위치시킬 DictionaryApp 클래스를 생성하고, 문제에 주어진 바와 같이 작성한다.
2. Dictionary 클래스를 생성하여 PairMap 을 상속받는다.
3. key 값과 value 값을 상속받아 저장하고 count 값을 1 증가한다.
4. 입력받은 key 가 있는지 검색하고, 찾지 못할 경우 null 을 리턴하고, 찾을 경우 찾은 key 값의 value 를 수정한다.
5. key 를 삭제할 경우, value 배열 중 같은 번호의 자리에 저장되어 있는 value 값 또한 삭제하고, 두 배열의 뒷 부분 데이터의 자리를 한 칸씩 앞으로 당긴 후 count 값을 1 감소한다.

#### C. 소스코드 및 주석

```
class Dictionary extends PairMap {
    protected int count = 0; // 저장된 값의 개수

    public Dictionary(int capacity) { // 생성자
        keyArray = new String [capacity];
        valueArray = new String [capacity];
    }

    @Override
    String get(String key) {
        for(int i=0; i<count; i++) { // 배열에 저장된 값의 개수만큼 반복
            if(keyArray[i].equals(key))
                return valueArray[i];
        }
    }
}
```

```

        }
        return null; // key를 발견할 수 없을 경우 null 리턴
    }

    @Override
    void put(String key, String value) {
        int i=0;
        for(i=0; i<count; i++) { // 배열에 저장된 값의 개수만큼 반복
            if(keyArray[i].equals(key))           break;
        }

        if(i == count) { // key를 발견하지 못할 경우
            if(i < keyArray.length) {
                keyArray[i] = key;
                valueArray[i] = value;
                count++;
            }
        } else { // value를 수정만 하는 경우
            keyArray[i] = key;
            valueArray[i] = value;
        }
    }

    @Override
    String delete(String key) {
        int i=0;
        for(i=0; i<count; i++) { // 현재 배열에 저장된 원소 개수만큼 반복
            if(keyArray[i].equals(key))
                break;
        }

        if(i==count) // 발견 안됨
            return null;

        String value = valueArray[i];

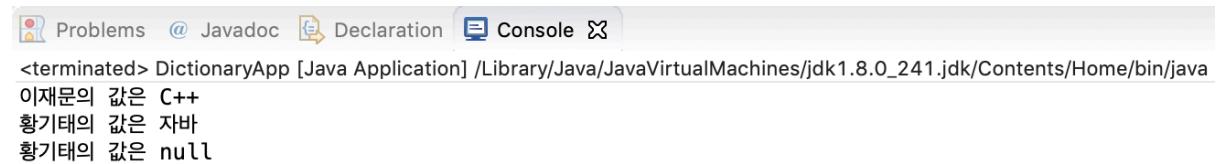
        // 앞으로 한 자리씩 이동
        int last = count-1;
        for(int j=i; j<last; j++) {
            keyArray[j] = keyArray[j+1];
            valueArray[j] = valueArray[j+1];
        }

        count--;
        return value;
    }

    @Override
    int length() { return count; }
}

```

## D. 입력 및 출력 결과 예시



The screenshot shows an IDE interface with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the following output:

```

<terminated> DictionaryApp [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_241.jdk/Contents/Home/bin/java
이재문의 값은 C++
황기태의 값은 자바
황기태의 값은 null

```

## 4. 인터페이스의 활용

### A. 문제

다음 도형 구성을 묘사하는 인터페이스이다.

```
interface Shape {
    final double PI = 3.14;
    void draw(); // 도형을 그리는 추상 메소드
    double getArea(); // 도형의 면적을 리턴하는 추상 메소드
    default public void redraw() { // 디폴트 메소드
        System.out.print("--- 다시 그립니다. ");
        draw();
    }
}
```

다음 main()메소드의 실행결과를 참고하여 인터페이스 Shape 을 구현한 클래스 Circle 을 작성하고, 전체 프로그램을 완성하시오.

```
public class Shapes {
    public static void main(String [] args) {
        Shape donut = new Circle(10); // 반지름이 10 인 원 객체
        donut.redraw();
        System.out.println("면적은 " + donut.getArea());
    }
}
```

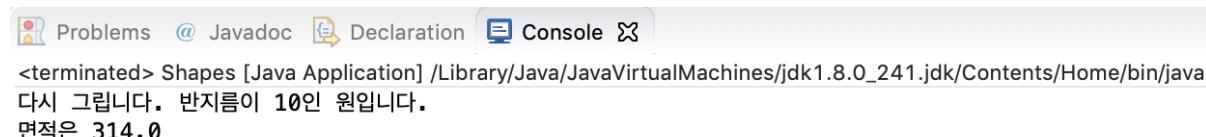
### B. 해결 방법

1. Shape 인터페이스를 상속받는 Circle 클래스를 생성한다.
2. 메인 메소드에서 지정하는 radius 값을 this. 를 통하여 불러온다.
3. radius 값을 출력하고 면적을 구하는 메소드를 작성하여 리턴한다.

### C. 소스코드 및 주석

```
public class Circle implements Shape { // Shape 인터페이스를 상속받음
    private int radius;
    public Circle(int radius) {
        this.radius = radius; // 메인 메소드에서 받게될 값
    }
    public void draw() {
        System.out.println("반지름이 "+radius+"인 원입니다.");
    }
    public double getArea() { // 면적 구하기
        return PI*radius*radius; // Shape 인터페이스에 PI값이 저장되어있으므로 그대로 상속받음
    }
}
```

### D. 입력 및 출력 결과 예시



The screenshot shows an IDE interface with tabs for Problems, Javadoc, Declaration, Console, and a close button. The Console tab is active, displaying the following text:

```
<terminated> Shapes [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_241.jdk/Contents/Home/bin/java
다시 그립니다. 반지름이 10인 원입니다.
면적은 314.0
```