

## JAVA 프로그래밍 및 실습

인공지능학부 214499 나유경

1.

```
proj1.java ×
5   public int getX() {return x;}
6   public int getY() {return y;}
7   protected void move(int x, int y) {this.x=x; this.y=y;}
8   }
9
10  class PositivePoint extends Point{ //point클래스를 상속받은 PositivePoint 클래스
11
12      @Override //오버라이드 확인
13      protected void move(int x, int y) { //move 메소드
14          if(x>0 && y>0) {
15              super.move(x, y); //부모클래스의 x, y 가져옴
16          }
17      }
18      public PositivePoint() { //PositivePoint 메서드
19          super(0,0); //부모클래스의 x,y를 0,0으로 가져옴(오버라이딩)
20      }
21      public PositivePoint(int x, int y) {
22          super(x, y); //부모클래스의 x,y를 그대로 가져옴-정적바인딩
23          if(x<=0 || y <= 0) { //x, y가 음수라면
24              super.move(0, 0); //x,y 값을 0으로 반환함
25          }
26      }
27      public String toString() { //toString 메서드, 자바에는 따로 String이 없을
28          String result = "("+getX()+","+getY()+")의 점";
29          return result;
30      }
31  }
32  }
33
34  public class proj1 { //문제 제시됨 메인 메소드
35      public static void main(String[] args) {
36
37          PositivePoint p = new PositivePoint(); //생성자PositivePoint에 객체p 생성
38          p.move(10,10); //동적바인딩 되어 10,10 반환됨
39          System.out.println(p.toString()+"입니다.");
40
41          p.move(-5,5); //객체 p는 음수 공간으로 이동되지 않음
42          System.out.println(p.toString()+"입니다.");
43
44          PositivePoint p2 = new PositivePoint(-10, -10); //-10은 음수이므로 0으로 반환됨
45          System.out.println(p2.toString()+"입니다.");
46      }
47  }
48  }
```

Problems @ Javadoc Declaration Console ×

<terminated> proj1 [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (2022. 5. 27. 오전 4:09:06 – 오전 4:09:06) [pid: 23048]

(10,10)의 점입니다.  
(10,10)의 점입니다.  
(0,0)의 점입니다.

2.

```
1 import java.util.Scanner;
2 interface Stack{ //문제에서 제시된 코드
3     int length(); //현재 스택에 저장된 개수 리턴
4     int capacity(); //스택의 전체 저장 가능한 개수 리턴
5     String pop(); //스택의 톱에 실수 저장
6     boolean push(String val); //스택의 톱에 저장된 실수 리턴
7 }
8
9 class StringStack implements Stack{ //인터페이스 구현
10     public int num; //스택에 저장 공간 크기
11     public int index; //저장 인덱스
12     public String[] stack;
13     public StringStack(int num) { //StringStack 메서드
14         this.num = num; //필드 변수 받아옴
15         this.index = index;
16         stack = new String[num];
17     }
18     public int length() { //length 메서드, 인터페이스에 제시된 메서드 코드 구현
19         return index; //저장 인덱스를 리턴함
20     }
21     public int capacity() { //capacity 메서드, 위와 동일
22         return stack.length; //스택의 저장 인덱스 리턴
23     }
24     public String pop() {
25         if(index<0) //Stack 에 아무것도 안들어왔을 때
26             return null;
27         index--; //스택의 인덱스값을 내려줌
28         String str = stack[index]; // 스트링 변수
29         return str;
30     }
31     public boolean push(String val) {
32         if (index<num) { //스택 최상위 값(맨 마지막에 들어온 값)의 인덱스가 스택의 크기보다 작을 때
33             stack[index]=val;
34             index++; //stack 최상위 값의 인덱스 올려줌
35             return true;
36         }
37         else
38             return false; //스택이 다 찼으면 false값 리턴
39     }
40 }
41
42 public class proj2 { //StackApp클래스 명칭 대신에 proj2 이름 씀
43     public static void main(String[] args) {
44         System.out.print("총 스택 저장 공간의 크기 입력 >> ");
45         Scanner sc = new Scanner(System.in); //스캔
46         int num = sc.nextInt(); //스캔 값 num에 받아옴
47         StringStack stack = new StringStack(num);
48         while(true) { //무한반복
49             System.out.print("문자열 입력 >> ");
50             String val = sc.next();
51             if(val.equals("그만")) //그만이라는 값이 입력되었으면
52                 break;
53             if(!stack.push(val)) { //스택 푸시에 실패했을 시
54                 System.out.println("스택이 꽉차서 푸시 불가!");
55             }
56         }
57         System.out.print("스택에 저장된 모든 문자열 팔 : ");
58         int len = stack.length(); //for문에서 바로 우항 입력하면 한개 줄림
59         for(int i=0; i<len; i++) { //스택의 길이만큼 반복, i<=len으로 하면 null값까지 출력
60             String str = stack.pop(); //stack 변수에서 stack의 최상위값 pop
61             System.out.print(str+ " "); //공백으로 문자 구분
62         }
63     }
64 }
65
```

Console ×

<terminated> proj2 [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (2022. 5. 27. 오후 12:24:45 - 오후 12:24:45)

총 스택 저장 공간의 크기 입력 >> 3

문자열 입력 >> hello

문자열 입력 >> sunny

문자열 입력 >> smile

문자열 입력 >> happy

스택이 꽉차서 푸시 불가!

문자열 입력 >> 그만

스택에 저장된 모든 문자열 팔 : smile sunny hello

3.

```

1 abstract class PairMap{
2     protected String keyArray[]; //key 들을 저장하는 배열
3     protected String valueArray[]; //value 들을 저장하는 배열
4     abstract String get(String key);
5     abstract void put(String key, String value); //key와 value를 쌍으로 저장
6
7     abstract String delete(String key); //key값을 가진 아이템(value와 함께) 삭제. 삭제된 value 값 리턴
8     abstract int length(); //현재 저장된 아이템의 개수 리턴
9 }
10
11 class Dictionary extends PairMap{ //PairMap 상속받은 Dictionary 클래스
12     public int size; //담을 수 있는 데이터 크기
13     public Dictionary(int num) { //디렉터리 클래스
14         keyArray = new String[num];
15         valueArray = new String[num];
16         this.size = 0; //필드 변수 size를 0으로 반환함
17     }
18
19     @Override
20     public String get(String key) {
21         for(int i=0; i<keyArray.length; i++) { //배열의 크기만큼 반복
22             if(key.equals(keyArray[i])) { //key값의 i에 값이 같으면
23                 return valueArray[i]; //Value 값 반환
24             }
25         }
26         return null; // 원하는 key가 없다면 null 리턴
27     }
28     public void put(String key, String value) { //put 메서드
29         for(int i=0; i<keyArray.length; i++) { //key 배열 만큼 반복
30             if(key.equals(keyArray[i])) {
31                 keyArray[i] = key;
32                 valueArray[i] = value;
33                 return; // 중복 key가 있다면 종료
34             }
35         }
36         keyArray[size] = key;
37         valueArray[size] = value;
38         size++;
39     }
40     public String delete(String key) {
41         for(int i=0; i<keyArray.length; i++) {
42             if(key.equals(keyArray[i])) {
43                 String str = valueArray[i]; //str 변수에는 value값 저장
44                 keyArray[i] = null;
45                 valueArray[i] = null;
46                 return str; // 삭제된 value 값 리턴
47             }
48         }
49         return null; // 삭제된 것이 없다면 null 리턴
50     }
51     public int length() {
52         return size;
53     }
54 }
55
56
57 public class proj3 { //DictionaryApp 명칭 proj3으로 바꿈
58     public static void main (String[] args){
59         Dictionary dic = new Dictionary(10);
60         dic.put("황기태", "자바");
61         dic.put("이재문", "파이썬");
62         dic.put("이재문", "C++"); // 이재문의 값을 C++로 수정
63         System.out.println("이재문의 값은 "+ dic.get("이재문")); // 이재문 아이템 출력
64         System.out.println("황기태의 값은 "+ dic.get("황기태")); // 황기태 아이템 출력
65         dic.delete("황기태"); // 황기태 아이템 삭제
66         System.out.println("황기태의 값은 "+ dic.get("황기태")); // 삭제된 아이템 접근
67     }
68 }
69

```

Console X

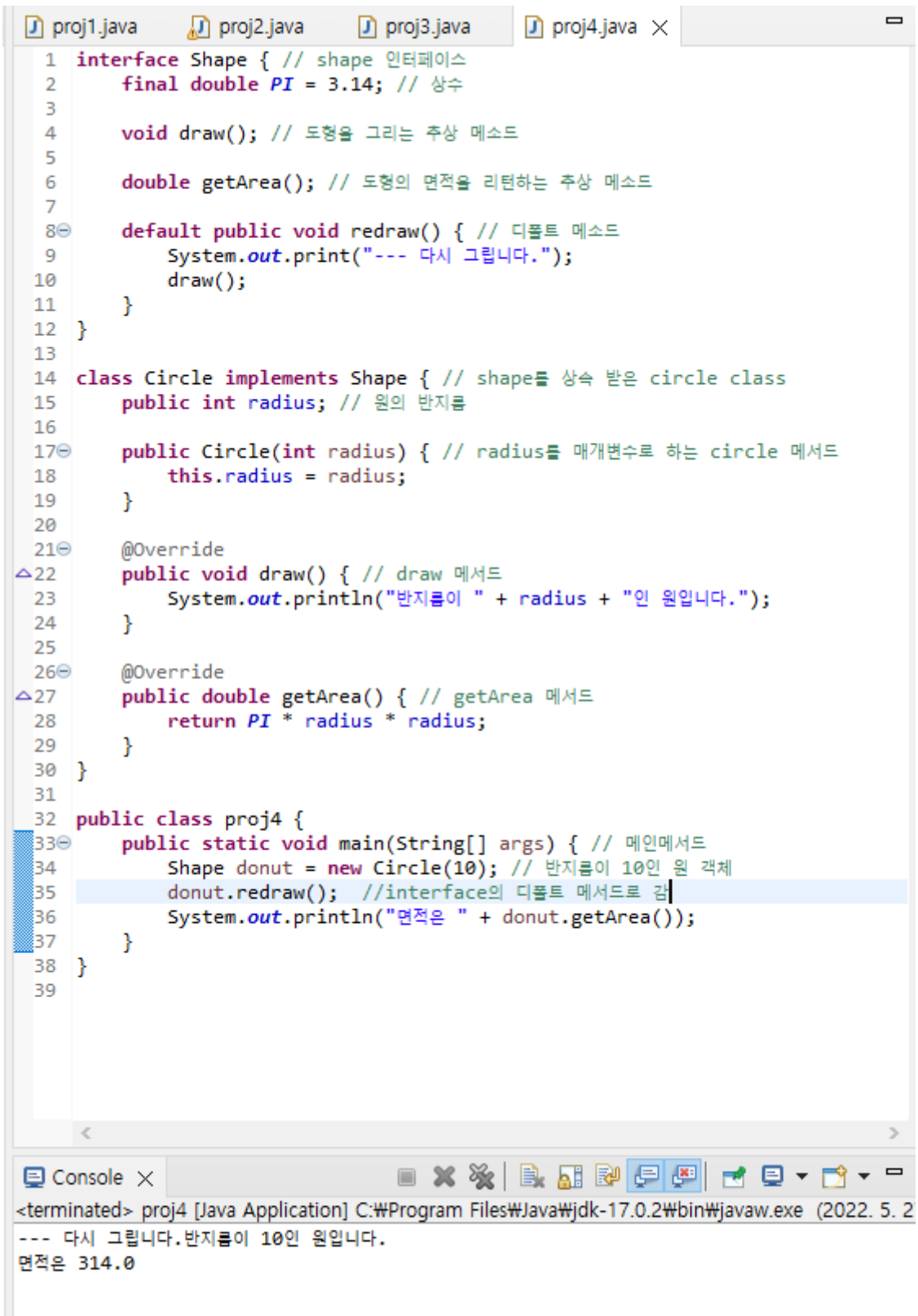
<terminated> proj3 [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (2022. 5. 27. 오후 12:51:20 - S

이재문의 값은 C++

황기태의 값은 null

황기태의 값은 null

4.



```
1 interface Shape { // shape 인터페이스
2     final double PI = 3.14; // 상수
3
4     void draw(); // 도형을 그리는 추상 메소드
5
6     double getArea(); // 도형의 면적을 리턴하는 추상 메소드
7
8     default public void redraw() { // 디폴트 메소드
9         System.out.print("--- 다시 그립니다.");
10        draw();
11    }
12 }
13
14 class Circle implements Shape { // shape를 상속 받은 circle class
15     public int radius; // 원의 반지름
16
17     public Circle(int radius) { // radius를 매개변수로 하는 circle 메서드
18         this.radius = radius;
19     }
20
21     @Override
22     public void draw() { // draw 메서드
23         System.out.println("반지름이 " + radius + "인 원입니다.");
24     }
25
26     @Override
27     public double getArea() { // getArea 메서드
28         return PI * radius * radius;
29     }
30 }
31
32 public class proj4 {
33     public static void main(String[] args) { // 메인메서드
34         Shape donut = new Circle(10); // 반지름이 10인 원 객체
35         donut.redraw(); //interface의 디폴트 메서드로 감
36         System.out.println("면적은 " + donut.getArea());
37     }
38 }
39
```

Console ×

<terminated> proj4 [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (2022. 5. 2  
--- 다시 그립니다.반지름이 10인 원입니다.  
면적은 314.0

5.

```

1 interface Shape2 { // shape 인터페이스 - proj4에 이미 정의되어있으므로 2라고 붙임
2     final double PI = 3.14; // 상수
3
4     void draw(); // 도형을 그리는 추상 메소드
5
6     double getArea(); // 도형의 면적을 리턴하는 추상 메소드
7
8     default public void redraw() { // 디폴트 메소드
9         System.out.print("--- 다시 그립니다.");
10        draw();
11    }
12 }
13
14 class Circle2 implements Shape2 { // shape를 상속 받은 circle class proj4에 이미 정의되었으므로 2라고 붙임
15     private int radius; // 원의 반지름, 접근지정자 private로 받아서 해당 클래스 내에서만 사용
16
17     public Circle2(int radius) { // radius를 매개변수로 하는 circle 메서드
18         this.radius = radius;
19     }
20
21     @Override
22     public void draw() { // draw 메서드
23         System.out.println("반지름이 " + radius + "인 원입니다.");
24     }
25
26     @Override
27     public double getArea() { // getArea 메서드
28         return PI * radius * radius;
29     }
30 }
31
32 class Oval implements Shape2 { //Shape2를 상속 받는 Oval 클래스
33     private int a, b;
34
35     public Oval(int a, int b) { //Oval 클래스
36         this.a = a; //필드 변수 a, b를 가져옴
37         this.b = b;
38     }
39
40     @Override
41     public void draw() { // draw 메서드
42         System.out.println(a + "x" + b + "에 내접하는 타원입니다.");
43     }
44
45     @Override
46     public double getArea() { // getArea 메서드
47         return PI * a * b;
48     }
49 }
50
51 class Rect implements Shape2 { //Shape2를 상속 받는 Rect 클래스
52     private int a, b;
53
54     public Rect(int a, int b) {
55         this.a = a; //필드 변수 a, b를 가져옴
56         this.b = b;
57     }
58
59     public void draw() { // draw 메서드
60         System.out.println(a + "x" + b + "크기의 사각형 입니다.");
61     }
62
63     public double getArea() { // getArea 메서드
64         return a * b;
65     }
66 }
67
68 public class proj5 {
69     public static void main(String[] args) {
70         Shape2[] list = new Shape2[3]; // Shape를 상속받은 클래스 객체의 레퍼런스 배열
71         list[0] = new Circle2(10); // 반지름이 10인 원 객체
72         list[1] = new Oval(20, 30); // 20x30 사각형에 내접하는 타원
73         list[2] = new Rect(10, 40); // 10x40 크기의 사각형
74         for (int i = 0; i < list.length; i++)
75             list[i].redraw();
76         for (int i = 0; i < list.length; i++)
77             System.out.println("면적은 " + list[i].getArea());
78     }
79 }
80

```

Console X

<terminated> proj5 [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (2022. 5. 27. 오후 2:00)

```

--- 다시 그립니다.반지름이 10인 원입니다.
--- 다시 그립니다.20x30에 내접하는 타원입니다.
--- 다시 그립니다.10x40크기의 사각형 입니다.
면적은 314.0
면적은 1884.0000000000002
면적은 400.0

```