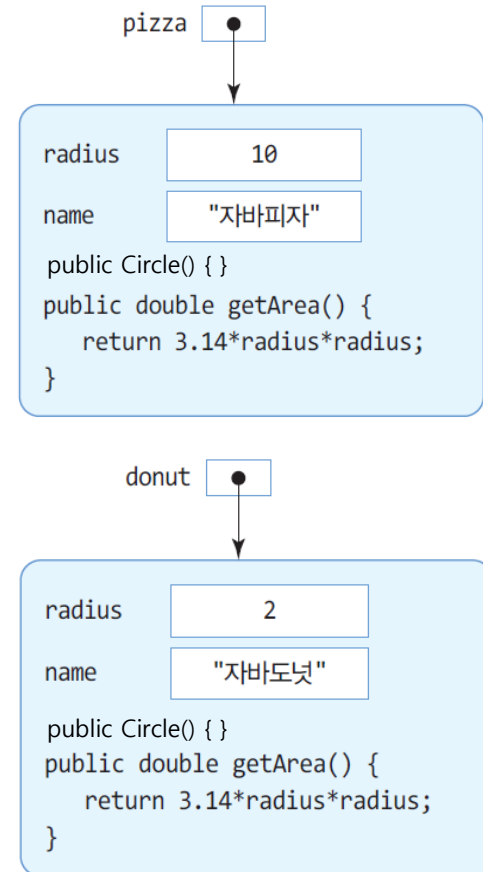


예제 4-1 : Circle 클래스의 객체 생성 및 활용

1

```
public class Circle {  
    int radius;           // 원의 반지름 필드  
    String name;          // 원의 이름 필드  
  
    public Circle() { }   // 원의 생성자  
  
    public double getArea() { // 원의 면적 계산 메소드  
        return 3.14*radius*radius;  
    }  
  
    public static void main(String[] args) {  
        Circle pizza;  
        pizza = new Circle();           // Circle 객체 생성  
        pizza.radius = 10;              // 피자의 반지름을 10으로 설정  
        pizza.name = "자바피자";        // 피자의 이름 설정  
        double area = pizza.getArea();  // 피자의 면적 알아내기  
        System.out.println(pizza.name + "의 면적은 " + area);  
  
        Circle donut = new Circle();    // Circle 객체 생성  
        donut.radius = 2;               // 도넛의 반지름을 2로 설정  
        donut.name = "자바도넛";         // 도넛의 이름 설정  
        area = donut.getArea();          // 도넛의 면적 알아내기  
        System.out.println(donut.name + "의 면적은 " + area);  
    }  
}
```



- ➔ 본인의 학번과 이름을 먼저 출력하고 프로그램을 진행하도록 함
- ➔ 자바 도넛 대신 본인의 이름과 번호(뒷 2자리)로 이름과 반지름 값을 주어 객체를 생성하고 값 출력 추가

예제 4-2 : Rectangle 클래스 만들기 연습

2

너비와 높이를 입력 받아 사각형의 합을 출력하는 프로그램을 작성하라. 너비(width)와 높이(height) 필드, 그리고 면적 값을 제공하는 getArea() 메소드를 가진 Rectangle 클래스를 만들어 활용하라.

```
import java.util.Scanner;

public class Rectangle {
    int width;
    int height;

    public int getArea() {
        return width*height;
    }

    public static void main(String[] args) {
        Rectangle rect = new Rectangle(); // 객체 생성
        Scanner scanner = new Scanner(System.in);
        System.out.print("> > ");

        rect.width = scanner.nextInt();
        rect.height = scanner.nextInt();

        System.out.println("사각형의 면적은 " + rect.getArea());

        scanner.close();
    }
}
```

➔ 위 프로그램을 정사각형으로 변경하여 입력 전에 한 변의 길이를 입력하라는 메시지를 출력한 후, 변의 길이를 입력 받아 면적을 계산할 수 있도록 변환하시오.

예제 4-3 : 두 개의 생성자를 가진 Circle 클래스

3

다음 코드는 2개의 생성자를 가진 Circle 클래스이다. 예제 4-1에서와 같이 본인의 이름과 번호를 이용하여 Circle을 생성하도록 수정하시오

```
public class Circle {
    int radius;
    String name;

    public Circle() { // 매개 변수 없는 생성자
        radius = 1; name = ""; // radius의 초기값은 1
    }
    public Circle(int r, String n) { // 매개 변수를 가진 생성자
        radius = r; name = n;
    }
    public double getArea() {
        return 3.14*radius*radius;
    }

    public static void main(String[] args) {
        Circle pizza = new Circle(10, "자바피자"); // Circle 객체 생성, 반지름 10

        double area = pizza.getArea();
        System.out.println(pizza.name + "의 면적은 " + area);

        Circle donut = new Circle(); // Circle 객체 생성, 반지름 1
        donut.name = "도넛피자";
        area = donut.getArea();
        System.out.println(donut.name + "의 면적은 " + area);
    }
}
```

예제 4-5 this()로 다른 생성자 호출

4

```
public class Book {
    String title;
    String author;
    void show() { System.out.println(title + " " + author); }

    public Book() {
        this("", "");
        System.out.println("생성자 호출됨");
    }

    public Book(String title) {
        this(title, "작자미상");
    }

    public Book(String title, String author) {
        this.title = title; this.author = author;
    }

    public static void main(String [] args) {
        Book littlePrince = new Book("어린왕자", "생텍쥐페리");
        Book loveStory = new Book("춘향전");
        Book emptyBook = new Book();
        loveStory.show();

        littlePrince.show();
    }
}
```

- ➔ 예제 4-4에서 작성한 Book 클래스의 생성자를 this()를 이용하여 수정하시오.
- ➔ main에서 객체 생성 시 하나는 본인의 이름을 저자로 하고 교재 제목을 책 제목으로 객체를 생성하고, 또 하나는 "자바 " 라는 제목만 가지고 객체를 생성하여 그 결과를 출력하시오