

1. 다음의 요구를 만족하는 추상 클래스 Calculator를 작성하고, Calculator를 상속받는 AddCalculator 클래스를 작성하라. AddCalculator 클래스는 아래의 main() 메서드를 이용해 테스트하라.

- 정수형 필드 num1, num2가 있다.
- num1, num2를 사용하여 특정 계산을 수행하는 int calc() 메서드를 추상 메서드로 정의한다.
- void prtResult() 메서드는 calc() 메서드 실행 결과를 리턴 받아 출력하도록 구현한다.
- 단, Calculator 클래스의 모든 멤버는 서브 클래스에만 접근을 허용한다.

<테스트를 위한 main() 메서드>

```
public static void main(String[] args) {  
    Calculator calc = new AddCalculator(10, 20);  
    calc.prtResult();  
}
```

<실행결과>

실행 결과는 30 입니다.

```
public abstract class Calculator {  
    protected int num1, num2;  
  
    public Calculator(int num1, int num2) {  
        this.num1 = num1;  
        this.num2 = num2;  
    }  
  
    protected abstract int calc();  
  
    protected void prtResult() {  
        System.out.printf("실행 결과는 %d 입니다.\n", calc());  
    }  
}
```

```
public class AddCalculator extends Calculator {  
  
    public AddCalculator(int num1, int num2) {  
        super(num1, num2);  
    }  
  
    @Override  
    protected int calc() {  
        return num1 + num2;  
    }  
}
```

```
public class TestCalculator {  
  
    public static void main(String[] args) {  
        Calculator calc = new AddCalculator(10, 20);  
        calc.prtResult();  
    }  
}
```

2. 다음은 도형의 구성을 묘사하는 인터페이스이다.

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

다음 main() 메서드와 실행 결과를 참고하여, Shape을 구현하는 클래스 Circle을 작성하고 프로그램을 완성하라.

< main() 메서드 >

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

<실행 결과>

```
--- 다시 그립니다. 반지름이 10인 원입니다.
면적은 314.0
```

```
public class Circle implements Shape {

    private int radius;

    public Circle(int radius) {
        this.radius = radius;
    }

    @Override
    public void draw() {
        System.out.printf("반지름이 %d인 원입니다.\n", radius);
    }

    @Override
    public double getArea() {
        return PI * radius * radius;
    }

}
```

```
public class TestCircle {

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

}
```

3. 실행 결과를 참고하여 2번 문제의 Shape 인터페이스를 구현한 클래스 Oval, Rect를 추가 작성하고 아래의 main() 메서드를 완성하라.(단, 타원의 면적은 'PI * 너비 * 높이' 이다.)

< main() 메서드 >

```
public static void main(String[] args) {  
    Shape[] list = {new Circle(10), new Oval(20, 30), new Rect(10, 40)};  
  
    for(Shape s : list) {  
        s.redraw();  
        System.out.printf("면적은 %.2f\n", s.getArea());  
    }  
}
```

<실행 결과>

```
--- 다시 그립니다. 반지름이 10인 원입니다.  
면적은 314.00  
--- 다시 그립니다. 20x30에 내접하는 타원 입니다.  
면적은 1884.00  
--- 다시 그립니다. 10x40 크기의 사각형입니다.  
면적은 400.00
```

```
public class Oval implements Shape {  
  
    private int width, height;  
  
    public Oval(int width, int height) {  
        this.width = width;  
        this.height = height;  
    }  
  
    @Override  
    public void draw() {  
        System.out.printf("%dx%d에 내접하는 타원 입니다.\n", width, height);  
    }  
  
    @Override  
    public double getArea() {  
        return PI * width * height;  
    }  
}
```

```
public class Rect implements Shape {

    private int width, height;

    public Rect(int width, int height) {
        this.width = width;
        this.height = height;
    }

    @Override
    public void draw() {
        System.out.printf("%dx%d 크기의 사각형입니다.\n", width, height);
    }

    @Override
    public double getArea() {
        return width * height;
    }

}
```

```
public class TestShape {

    public static void main(String[] args) {
        Shape[] list = {new Circle(10), new Oval(20, 30), new Rect(10, 40)};

        for(Shape s : list) {
            s.redraw();
            System.out.printf("면적은 %.2f\n", s.getArea());
        }

    }

}
```