

The background is a dark blue gradient with a subtle pattern of small white dots. Overlaid on this are several faint, light blue circular elements. On the left side, there are concentric circles with degree markings ranging from 40 to 260. Some of these circles have arrows indicating a clockwise direction. Other smaller circular patterns are scattered across the upper and lower portions of the image.

# Observer 패턴 실습 문제

# 요구 사항

- 숫자를 생성하는 Subject와 생성된 숫자를 통지 받아 서로 다른 방식으로 출력하는 Observer들
- NumberGenerator 추상클래스
  - Subject를 위한 인터페이스
  - 관찰자 등록, 해지, 통지를 위한 메소드 선언
  - 번호를 생성하라고 요청할 때 호출하는 메소드 선언(execute( ))
  - 관찰자가 번호를 얻어갈 때 호출하는 메소드 선언(getNumber())
  - 관찰자 리스트를 유지하기 위한 속성 선언 및 생성(observers)
- RandomNumberGenerator 클래스
  - NumberGenerator 인터페이스를 구현
  - execute( ): 0~99 사이의 임의의 정수를 생성하고 관찰자에게 통지
    - java.util.Random 클래스 이용(Random 객체의 nextInt(100) 호출)

# 요구 사항

- Observer 인터페이스
  - 관찰자를 위한 인터페이스
  - 통지 받을 때 호출되는 메소드 선언(update( ))
- DigitObserver 클래스
  - 구체적인 관찰자
  - update( ):
    - Subject가 생성한 숫자를 얻어서 그대로 출력
- GraphObserver 클래스
  - 구체적인 관찰자
  - update( ):
    - Subject가 생성한 숫자를 얻어서 그 숫자만큼 '\*' 문자를 출력

# 요구 사항

- Client 클래스
  - 구체적인 Subject 생성
  - 구체적인 관찰자 2종류 생성
  - 관찰 대상에게 관찰자 등록
  - 20번 반복: 관찰대상의 execute( ) 호출하여 임의의 수 생성하기
    - 관찰자들에게 자동으로 통지가 감
  - 첫번째 관찰자 삭제
  - 20번 반복: 관찰대상의 execute( ) 호출하여 임의의 수 생성하기



# 출력 예

```
DigitObserver:9
GraphObserver:*****
DigitObserver:32
GraphObserver:*****
DigitObserver:22
GraphObserver:*****
DigitObserver:48
GraphObserver:*****
DigitObserver:35
GraphObserver:*****
DigitObserver:18
```

(20회 출력....)

# 클래스 다이어그램

