KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

Observer - 상태 변화를 알려 준다

[KB] IT's Your Life

KB 국민은행

Observer 패턴

- 관찰 대상의 상태가 변화하면 관찰자에게 알림
- 상태 변화에 따른 처리를 기술할 때 효과적

Observer 패턴 예제 프로그램

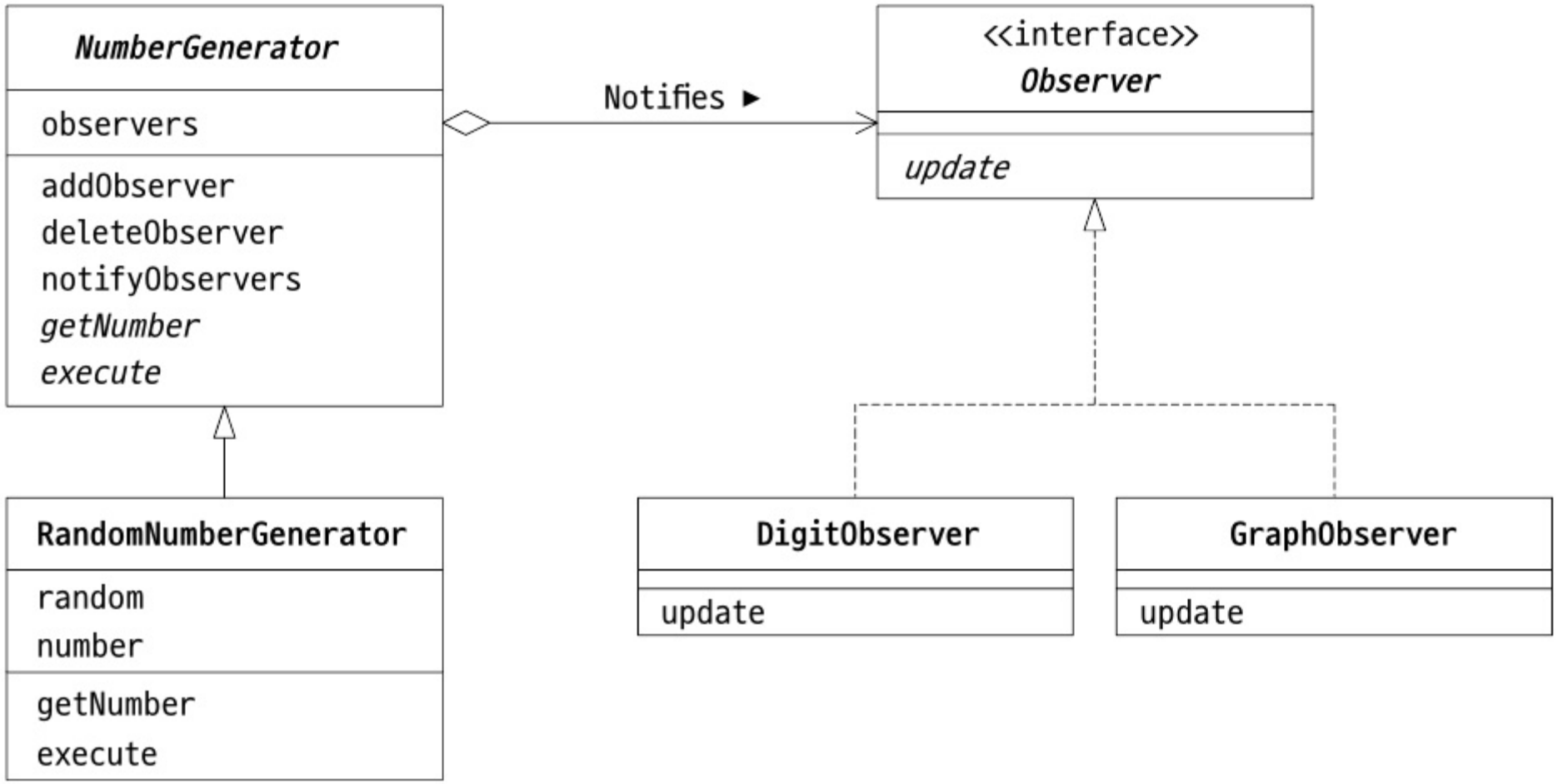
예제 프로그램

- 수를 많이 생성하는 객체를 관찰자가 관찰하고 그 값을 표시하는 프로그램
- 표시하는 방법은 관찰자에 따라 다름
  - DigitObserver: 숫자로 표시
  - GraphObserver: 간단한 그래프로 표시

이름	설명
Observer	관찰자를 나타내는 인터페이스
NumberGenerator	수를 생성하는 객체를 나타내는 추상 클래스
RandomNumberGenerator	랜덤하게 수를 생성하는 클래스
DigitObserver	숫자로 수를 표시하는 클래스
GraphObserver	간이 그래프로 수를 표시하는 클래스
Main	동작 테스트용 클래스

Observer 패턴 예제 프로그램

Observer 패턴의 클래스 다이어그램



Observer 패턴 예제 프로그램

Observer.java

```
public interface Observer {
    void update(NumberGenerator generator);
}
```

Observer 패턴 예제 프로그램

NumberGenerator.java

```
public abstract class NumberGenerator {
    // Observer를 저장한다.
    private List<Observer> observers = new ArrayList<>();

    // Observer를 추가한다.
    public void addObserver(Observer observer) {
        observers.add(observer);
    }

    // Observer를 제거한다.
    public void deleteObserver(Observer observer) {
        observers.remove(observer);
    }

    // Observer에 통지한다.
    public void notifyObservers() {
        for(Observer o: observers) {
            o.update(this);
        }
    }
```

Observer 패턴 예제 프로그램

NumberGenerator.java

```
    // 수를 취득한다.
    public abstract int getNumber();

    // 수를 생성한다.
    public abstract void execute();
}
```

Observer 패턴 예제 프로그램

RandomNumberGenerator.java

```
public class RandomNumberGenerator extends NumberGenerator {
  private Random random = new Random();  // 난수 생성기
  private int number;                    // 현재 수

  // 수를 취득한다
  @Override
  public int getNumber() {
    return number;
  }

  // 수를 생성한다
  @Override
  public void execute() {
    for (int i = 0; i < 20; i++) {
      number = random.nextInt(50);
      notifyObservers();
    }
  }
}
```

Observer 패턴 예제 프로그램

DigitObserver.java

```
public class DigitObserver implements Observer {
  @Override
  public void update(NumberGenerator generator) {
    System.out.println("DigitObserver:" + generator.getNumber());
    try {
      Thread.sleep(100);
    } catch (InterruptedException e) {
    }
  }
}
```

Observer 패턴 예제 프로그램

GraphObserver.java

```
public class GraphObserver implements Observer {
  @Override
  public void update(NumberGenerator generator) {
    System.out.print("GraphObserver:");

    int count = generator.getNumber();

    for (int i = 0; i < count; i++) {
      System.out.print("*");
    }
    System.out.println("");

    try {
      Thread.sleep(100);
    } catch (InterruptedException e) {
    }
  }
}
```

Observer 패턴 예제 프로그램

Main.java

```
public class Main {
  public static void main(String[] args) {
    NumberGenerator generator = new RandomNumberGenerator();

    Observer observer1 = new DigitObserver();
    Observer observer2 = new GraphObserver();

    generator.addObserver(observer1);
    generator.addObserver(observer2);

    generator.execute();
  }
}
```

```
DigitObserver:38
GraphObserver:**************************************
DigitObserver:39
GraphObserver:***************************************
DigitObserver:49
GraphObserver:************************************************
DigitObserver:7
GraphObserver:*******
DigitObserver:14
GraphObserver:**************
DigitObserver:26
GraphObserver:**************************
DigitObserver:11
GraphObserver:***********
DigitObserver:3
GraphObserver:***
...
DigitObserver:22
GraphObserver:**********************
DigitObserver:43
GraphObserver:*******************************************
DigitObserver:33
GraphObserver:********************************
```

Observer 패턴

Observer 패턴의 클래스 다이어그램