KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

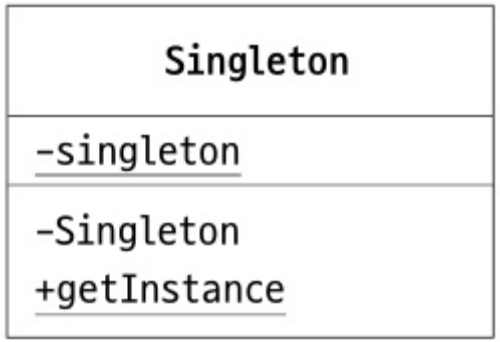
Singleton - 인스턴스를 단 하나만 만든다

[KB] IT's Your Life

KB 국민은행

Singleton 패턴

Singleton 패턴의 클래스 다이어그램



Singleton.java

```
public class Singleton {
    private static Singleton singleton = new Singleton();

    private Singleton() {
        System.out.println("인스턴스를 생성했습니다.");
    }

    public static Singleton getInstance() {
        return singleton;
    }
}
```

예제 프로그램

Main.java

```
public class Main {
    public static void main(String[] args) {
        System.out.println("Start.");

        Singleton obj1 = Singleton.getInstance();
        Singleton obj2 = Singleton.getInstance();

        if(obj1 == obj2) {
            System.out.println("obj1과 obj2는 같은 인스턴스입니다.");
        } else {
            System.out.println("obj1과 obj2는 같은 인스턴스가 아닙니다.");
        }

        System.out.println("End.");
    }
}
```

```
Start.
인스턴스를 생성했습니다.
obj1과 obj2는 같은 인스턴스입니다.
End.
```

예제 프로그램

싱글톤 패턴을 사용하는 이유

인스턴스의 수를 제한

- 인스턴스가 여러 개 존재하면 인스턴스가 서로 영향을 미침 → 버그의 가능성이 높아짐

enum을 이용한 Singleton

```
enum Singleton {
  INSTANCE;
  public void hello() {
     System.out.println("hello is called.");
  }
}

Singleton.INSTANCE.hello();
```