

3

리스코프 치환 원칙 (LSP - Liskov Substitution Principle)

정의

프로그램의 객체는 프로그램의 정확성을 깨뜨리지 않으면서 하위 타입의 인스턴스로 바꿀 수 있어야 한다. 계약에 의한 설계를 참고하라.

계약이란?



각 계약 당사자는 계약으로부터 이익을 기대하고 의무를 이행한다.
각 계약 당사자의 이익과 의무는 계약서(인터페이스)에 문서화 된다.

계약에 의한 설계란?



의도를 드러내는 인터페이스를 만들어서 협력을 수행하기 위해 수행해야 하는 제약조건을 명시한다.

계약에 의한 설계를 구성하는 세 가지 요소

1. 사전조건

- 메소드가 호출되기 위해 만족되어야 하는 조건, 메소드의 요구사항 명시, 만족시켜야 하는 의무

- 일반적으로 매개변수의 정합성을 체크하기 위해 사용하며, 만족시키지 못할 경우 예외 발생

2. 사후조건

- 메소드가 실행 후 클라이언트에 보장해야 하는 조건, 만족시켜야 하는 서버의 의무
- 인스턴스 변수의 상태가 올바른지 서술
- 메소드에 전달된 파라미터의 값이 올바르게 변경되었는지 서술
- 반환값이 올바른지 서술
- 한 메소드 안에서 return문이 여러 번 나오는 경우에는 검증이 힘들게 된다.

3. 불변식

- 보장되어야 하는 서버의 조건, 메소드를 실행하기 전이나 종료된 후에 항상 불변식을 만족하는 상태가 유지되어야 함
- 인스턴스 생명주기 전반에 걸쳐 지켜져야 하는 객체 내부의 상태 규칙을 명세한다.
- 모든 인스턴스가 생성된 후에 만족되어야 하며, 클래스에 정의된 모든 생성자는 불변식을 준수해야 한다.
- 쉽게 말해 생성자 내부에서 참조 자료형인 경우 null이 아닌지 체크하거나, 논리적으로 규칙 상 위배되는 값이 발생하는 경우 예외 발생

해석



계약에 의한 설계를 기반으로 프로그램의 정확성을 깨뜨리지 않아야 한다는 것이다.



또한 리스코프 치환 원칙은 특정 문제를 해결하기 위한 설계 원칙이 아닌 상속과 다형성에 관한 원칙을 제공한다. 반드시 IS-A 관계가 성립이 될 때에만 상속을 적용하며, 이러한 클래스들 간에는 다형성을 이용하여 동작을 할 시, 하위 타입의 인스턴스로 바뀌어서 동작해도 문제가 발생하지 않게 한다.



상속은 역할로 추상화한 것을 구체화 하기 위해 사용해야 한다. 즉, 동일한 메시지를 수신해서 응답을 한다는 말은 동일한 책임을 가지고 있는 것을 의미한다. 그것을 역할이라고 한다.



다운캐스팅을 지양하고 동적바인딩을 활용하라는 의미이다. 다운캐스팅을 이용하여 메시지를 전송한다는 의미는 특정 객체만 수행할 수 있는 내용이다. 따라서 상위타입에는 존재하지 않는 기능이기에 때문에 상위 타입을 구현한 모든 하위 타입들이 그 기능을 제공할 것이라는 보장이 없는 상태이다. 따라서 하위 타입을 교환해서 실행 시 runtime exception을 발생할 가능성이 높다는 의미이다. 따라서 instanceof 키워드 혹은 downcasting을 사용하는 것은 전형적인 LSP 위반의 징조이다.

Appendix



LSP는 OCP를 받쳐주는 상속과 다형성에 관한 원칙을 제공한다. 따라서 LSP가 위반되면 OCP도 위반된다는 의미를 가진다. 즉, LSP를 위반하면 subtype을 추가할 때 마다 변경사항이 발생할 수 밖에 없기 때문에 변경에 취약한 악취를 풍기는 코드라는 의미이다.