5957 객체지향 프로그래밍

60222117 이서현

목차

주제 : Object-Oriented Paradigm

1. Object-Oriented Paradigm

"Object-Oriented Paradigm이란, A world is viewed as a set of objects interacting with each other을 의미한다."[참고문헌-1번] 즉. 세상의 구성요소를 객체와 상호작용으로 해석하는 것이다. Class는 객체를 만들어내는 type이기 때문에, 실제 instance인 객체와 객체들 사이의 상호작용이 세상을 구성한다고 본다.

A. Object Interaction

Object Interaction은 message passing과 관련된 association을 의미한다. Message passing이란 객체의 상호작용을 의미한다. 이 과정에서 메시지와 함수를 1대1로 매칭시키는 message handler을 해야한다.

위의 message handler에서 함수를 호출할 때, 특정 객체의 내부를 알아야 명령을 할 수 있기 때문에, 커플링이 발생한다. 커플링은 두 개의 엔터티가

얼마나 자주 상호작용하는지를 의미하는데 이는 최소화되어야 한다. "Cohesion(응집성)"[참고문헌-2번]은 내부에 존재하는 통신을 의미하기 때문에 cohesion은 최대화되어야 한다. 결론적으로 객체와 객체 사이의 독립성을 최대화하되, 둘 사이의 통신 또한 최대화하는 것이 이상적이다.

따라서 message Handler로 mapping table을 이용한다. 객체가 특정 명령을 내렸을 때, 어떠한 행동을 해야 할지, 어떠한 함수를 호출해야 할 지에 대한 정보를 나타낸다. 예를 들어, 이벤트가 발생했을 때, 이 이벤트에 맞는 함수를 호출하는 프로그램을 event driven program이라고 칭한다.

B. Object Association

Object Association은 앞서 이야기한 Interaction과 달리, 객체 사이의 message passing이 아닌 서로 다른 두 객체에서 하나의 객체가 다른 객체의 메모리 주소를 알고, 이를 통해 메시지를 보낼 수 있음을 의미한다.

Object Association은 object aggregation hierarchy에서 찾을 수 있다. Aggregation hierarchy는 parent-child 구조로, 부모 객체가 자식 객체의 메모리 할당 및 해제를 결정하는 lifecycle management 구조이다. 이렇게 부모 객체 안에 자식 객체가 있는 경우, 자식 객체가 부모 자식 내부에 존재하기 때문에 자식 객체의 메모리 저장 주소를 알 수 있다. 따라서 이런 경우에 object association이 이루어진다.

Object Association에서 부모 객체와 자식 객체의 association의 정도가 강할수록 composition hierarchy, 약할수록 aggregation hierarchy라고 하지만 굳이 구분은 자주 하지 않는다.

2. Class Hierarchy

Class Hierarchy는 객체가 아닌 class사이의 구조를 나타낸다. 이는 inheritance, Generation/Specialization, Extends로 나눌 수 있다.

A. Inheritance

Inheritance는 상속을 의미한다. 특정 부모 class의 모든 구조와 행위적 특성을 받은 것을 상속이라고 한다.

B. Generation/Specification

Generation은 여러 객체들의 공통성을 모아 일반화하는 것을 의미한다.

일 반화는 구조, 행위적으로 공통적인 특성을 일반화할 수 있다. 반면에, 공통적이지 않은 고유한 특징을 갖는 것을 specification을 의미한다. 이는 확장과 비슷한 개념이다.

C. Extends

Extends는 확장을 의미한다. 부모 class로부터 상속을 받아 generation을 하고, 이 중 specification을 통해 고유한 구조, 행위를 정의한다. 다시 말해서, 부모 class로부터 대부분의 구조와 특성을 상속받고, 이에 고유한 구조와 행위를 추가하는 것이다.

참고문헌

1)" Object-Oriented Paradigm",네이버 위키백과, https://ko.wikipedia.org/wiki/%EA%B0%9D%EC%B2%B4_%EC%A7%80%ED%96%A5_%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D, 상단 1번째줄-2번째줄 인용

2) "cohesion",네이버 어학사전, https://en.dict.naver.com/#/search?query=cohesion, 상단 1번째 줄 인용