

5957 객체지향 프로그래밍

60222117 이서현

목차

- 1. UML
- 2. Architecture
 - A. Structural View
 - B. Behavioral View

주제 : Architecture

1. UML("unified modeling language"[참고문헌 1번- 인용])

UML이란, object oriented modeling을 위한 diagramming language이다. 즉, 본격적으로 개발을 시작하기 전, 개발할 프로그램의 의미를 쉽게 그림으로 표현하는 것을 의미한다. 쉽다는 것은 정확한 관점으로 정리되어 있다는 뜻이고, 반대로 어렵다는 것은 정확한 관점으로 정리되어 있지 않고 관점이 존재하지 않다는 것이다. 또한, 의미는 명사적으로 나와 제3자가 공통적으로 지칭하는 것, 동사적으로 특정 변화에 대해 동의하는 것을 의미한다. 다시 말해 UML은 개발할 프로그램에 대해 그림으로 정확한 기준으로 정리해 표현하는 것을 의미한다.

2. Architecture

Software program을 개발할 때에는 사용자의 요구사항을 정리해 program으로 만들어야 한다. 이때, 요구사항을 정리하는 데에 사용되는 것이 Architecture라고 한다. Architecture는 사용자의 요구사항 및 세상을 변화하는 것과 변화하지 않는 것으로 구분하는 것으로부터 시작한다. 변화하지 않는 것을 기준으로 구분하는 것을 Structure View, 변화하는 것을 기준으로 구분하는 것을 Behavioral View라고 한다. 세상은 명사로만 혹은 동사로만 구성이 불가능한 것처럼 Class 또한

Structure 혹은 Behavioral View 중 하나만 사용해서는 완벽하게 정리할 수 없다. 따라서 Class를 작성할 때에는 Structure View, Behavioral View 두 관점 모두 적용해야 한다.

A. Structure View

Structure View란 변하지 않는 구조, 명사를 척도로 Class를 정리하는 것이다. 구조적으로 정의할 때는 고유한 속성 혹은 구성요소를 통해 정의할 수 있다. 예를 들어 자동차와 자동차의 구성요소, 자동차와 자동차의 색 등의 기준으로 Class를 정의하는 것이다. 그러나 이러한 구성요소 및 속성이 작동하려면 어떻게 변화하는지에 대한 Behavioral view를 빼놓을 수 없다.

B. Behavioral View

Behavioral View란 변하는 것 즉, 행위를 척도로 Class를 정리하는 것이다. 즉, 동사로 Class 정확히 메소드를 정의하는 것인데, 이는 절차적 관점과 속성값의 관점으로 나눌 수 있다. 어떠한 과정을 통해 프로그램이 실행되는지에 따라 절차를 기준으로 메서드를 작성하는 것을 절차적 관점이라고 한다. 반면, 요소의 속성값의 변화를 기준으로 메서드를 작성하는 것을 속성값의 관점이라고 한다. 그러나 속성값의 변화 혹은 행위를 기준으로 Class를 정리하려면, 행위의 주어 및 속성의 주어가 필요하기 때문에 Structure View를 빼놓을 수 없다.

참고문헌

1)"UML", 네이버 지식 백과

<https://terms.naver.com/entry.naver?docId=1181048&cid=40942&categoryId=32838>, 1번

째줄 인용