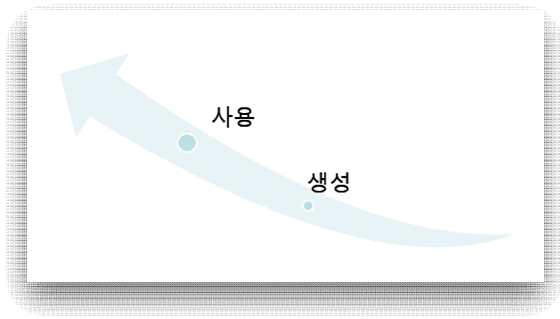


4장. 개념 I

(2023)



시작하기 전에

개념은 네 부분으로 나누어 설명됩니다. 이번 장은 개념의 첫 번째 부분으로 2장에서 설명된 실체에 대응하는 개념들을 아는 것을 목표로 합니다. 또한 생성과 사용이라는 개념에 대해서 분명히 알고 설명할 수 있어야 합니다.

이 장에서 사용되는 용어들은 다음과 같이 표기됩니다.

우리는 attribute, association, primitive type, enumeration을 외국어로 간주하고, 각각을 속성, 연관, 기본타입, 열거형으로 표기합니다.

우리는 class, data type, operation을 외래어로 간주하고, 각각을 클래스, 데이터타입, 오퍼레이션으로 표기합니다.

우리는 개념이라는 용어에 concept을 사용하지 않고 type을 사용합니다.

1절. 개념

1.1 개념(type)이 있어야 실체에 대해서 말할 수 있습니다.

실체에 대해서 말하려면 개념이 있어야 합니다. 우리는 객체, 값, 링크와 같은 실체들을 다루었습니다. 이들에 대해서 뭔가 말하려면 실체에 대응하는 개념이 있어야 합니다.

강의실에 있는 어떤 책상에 대해서 이야기 하려면 책상이라는 개념이 있어야 합니다. 만약 책상이라는 개념이 없다면 ‘저거 있잖아, 저거 말이야’와 같이 말할 뿐 책상에 대해서 말할 수 없습니다.

1.2 데이터타입(data type)은 값의 개념입니다.

1.3 UML에서는 데이터타입을 값의 형태에 따라 기본형(primitive), 열거형(enumeration), 데이터타입으로 구분합니다.

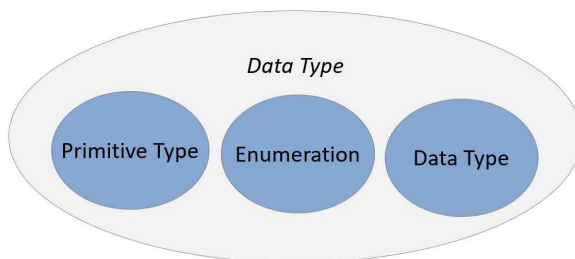


그림 4.1

기본형은 언어 자체에 정의된 타입으로, 사용자가 직접 정의하지 않고 사용할 수 있는 타입입니다.

UML은 기본형으로 Boolean, Integer, UnlimitedNatural, String, Real을 제공합니다.

고객구분이나 요일과 같이 사용할 수 있는 값들의 목록이 열거 가능한 수준으로 정해져 있을 때, 그 목록 중 하나를 사용하는 값의 개념을 열거형이라 합니다.

그룹값에 해당하는 데이터값의 개념을 데이터타입이라 합니다.

UML은 값의 개념과 데이터타입의 개념에 같은 이름인 '데이터타입'을 사용하고 있어 좀 혼란스러울 수 있습니다. 우리는 값의 개념을 이야기할 때는 보통 '값타입'이라고 합니다.

1.4 클래스(class)는 객체의 개념이고, 속성(attribute)은 특성의 개념입니다.

속성은 9장에서 좀 더 자세히 설명됩니다.

1.5 정체성에 대응하는 개념은 없습니다.

객체의 개념인 클래스는 정체성에 대응하는 개념은 제공하지 않습니다.

정체성은 객체 생성 시 정체성의 의미에 부합하도록 하는 특별한 메커니즘에 의해 부여됩니다.

정체성은 6장 개념 III과 9장 속성에서 좀 더 자세하게 설명됩니다.

1.6 연관(association)은 링크의 개념입니다.

링크의 역할쌍은 연관의 역할쌍으로 대응됩니다.

1.7 행위의 개념은 행위입니다.

클래스의 행위를 오퍼레이션(operation)이라고 합니다.

행위는 모든 객체에 대해 같은 방법으로 수행되도록 정의되기 때문에 실체에서는 큰 의미가 없습니다. 따라서 실체와 개념에 별도의 용어를 가져야하는 필요성이 적습니다.

우리는 전체와 부분의 개념을 갖고 객체를 다양한 수준으로 구분합니다. 객체의 구분에 따라 객체의 개념 또한 구분됩니다. 보통은 시스템, 서브시스템, 컴포넌트, 클래스로 구분합니다.

객체의 개념이 시스템이나 서브시스템일 경우, 행위는 목적행위 또는 유스케이스라 하고, 컴포넌트나 클래스일 경우 오퍼레이션이라고 합니다.

전체와 부분에 대해서는 6장 개념 III에서, 행위에 대해서는 10장, 11장에서 좀 더 자세히 설명됩니다.

2절. 생성과 사용

2.1 무리가 형성되면 개념이 생기고, 개념이 생기면 이름이 붙여지고 집합이 만들어집니다.

자연에 등장한 인류는 셀 수 없는 무수히 많은 객체들과 맞닥뜨리게 됩니다. 생존을 위해서는 이것들에 대해서 알고 서로 알고 있는 것을 교환해야 했습니다.

개개의 모든 객체들을 다 이해하고 그것에 대해 이야기하는 것은 불가능합니다. 여기서 놀라운 인간의 추상화 능력이 발휘됩니다.

자신들의 목적에 따라 개개의 객체들을 비교하면서 비슷하다고 생각하는 것들은 하나로 모으고 다르다고 생각하는 것들은 가릅니다. 생김새와 쓰임새가 기준이 됩니다. 그렇게 모음과 가름에 따라 무리가 이루어 집니다.

무리가 만들어졌다는 것은 개개의 객체들을 싸잡아 가리킬 수 있는 공통의 개념이 만들어졌다는 것입니다. 무리에 속한 모든 객체들에 대해서 '이것들은 이런 점에서 같고, 다른 것들과는 이런 점에서 달라'라고 말할 수 있게 된 것입니다.

개별 객체에만 적용되는 특징이 아닌 무리 전체에 두루 적용될 수 있는 보편 개념이 만들어진 것입니다.

무리가 만들어졌다는 것은 어떤 것은 그 무리에 들어올 수 있고 어떤 것은 들어올 수 없음을 나타내는 경계(boundary)가 쳐졌다는 것입니다. 경계를 치리면 그 안에 들어 있는 것들은 어떤 것들인지, 그 안에 들어올 수 있는 것은 어떤 것들이어야 하는지에 대해 분명한 정의(definition)가 내려져야 합니다. 생김새와 쓰임새에 의해서 정의가 내려집니다.

공통의 개념이 만들어졌기는 하지만 아직 이것을 가지고 의사소통하기

는 어렵습니다. ‘이런 점에서 같고, 다른 것들과는 이런 점에서 달라’를 매번 말하는 것은 귀찮은 일입니다. 여기서 인간의 놀라운 언어적 능력이 발휘됩니다. 공통의 개념을 갖는 무리에 이름을 붙입니다.

경계에 대한 정의와 이름이 붙여진 무리를 집합이라고 합니다.

2.2 객체는 사용 관점으로 이해되고 개념화됩니다.

인류가 등장했을 때 이미 자연이 있었습니다.

자연은 인류에게 있어 생성의 대상이 아니라 사용의 대상이었습니다. 인류는 오랫동안 사용 관점만을 가지고 객체를 이해하고 개념화해왔습니다.

아이들과 함께 봄날 산에 올라가 나무들과 꽃들을 보면서, “이것들은 나무고, 이것들은 꽃이야”라고 말할 때 나무나 꽃은 인간이 만든 것이 아닙니다. 이미 있던 것입니다.

‘나무’나 ‘꽃’이라는 개념은 나무나 꽃을 만들 때(생성)의 개념이 아니라 인간들이 사용을 위해 만든 개념입니다.

2.3 객체는 생성 관점으로 이해되고 개념화됩니다.

인류는 자연에 있는 것을 넘어 새로운 것을 추구합니다. 사용을 넘어 생성의 개념으로 객체들을 바라보기 시작한 것입니다.

새로운 객체를 만들어낼 수 있는 생성의 개념이 필요하게 된 것입니다.

2.4 객체는 생성된 후에야 비로소 사용될 수 있습니다.

객체가 사용되려면 먼저 생성되어야 합니다.

자연을 대상으로 오랫동안 사용을 중심으로 개념의 탐을 쌓아온 인류에게 생성이란 개념은 너무나 당연한 것이 아니라 큰 도전거리가 됩니다.

2.5 사용개념은 생성개념에 영향을 주지 못합니다.

생성을 고려하지 않은 사용개념은 관찰의 결과일 뿐입니다.

책상이 있는데, 책상에서 누워 잠을 자고 그것을 침대라고 부른다고 해서 책상이 침대로 생성되는 것은 아닙니다. 책상 개념으로 만들어진 책상을 단지 사용하면서 사용의 개념으로 침대라고 부른 것일 뿐입니다.

그림 4.2와 같이 도형들이 있고 사용을 위해 세 개의 집합으로 구분했습니다. 그리고 그림 4.3과 같이 이름을 붙였습니다.

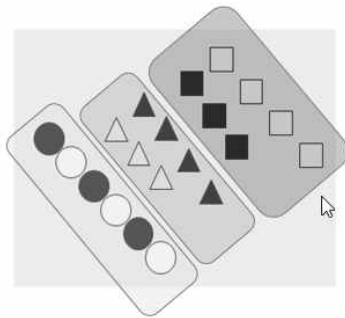


그림 4.2



4.3

이렇게 해도 원래의 원이나 사각형이나 삼각형의 모습에는 어떤 영향도 주지 못합니다. 그냥 부르는 이름이 달라진 것뿐입니다. 사용개념은 단지 사용을 위해 무리를 만들고 개념화해서 집합을 만든 것뿐입니다.

2.6 생성개념은 객체의 설계도입니다.

생성개념은 객체를 생성해 내는 개념입니다. 어떤 개념이 있고 도구와 재료가 있다면, 개념에 해당하는 객체를 만들 수 있습니다. 개념이 설계도가 되는 것입니다.

책상이라는 개념과 책상을 만들 도구와 재료가 있다면, 책상을 만들 수 있습니다. 책상이라는 개념이 설계도가 되어 (강의실에 있는 그) 책상 객체를 생성합니다.

2.7 생성개념도 집합을 형성합니다.

냉장고라는 개념을 통해 만들어진 객체들은 모두 냉장고입니다. 냉장고 개념을 통해 생성되는 모든 냉장고들은 냉장고라는 개념으로 묶을 수 있는 것입니다. 집합이 형성된 것입니다.

사용개념에 의한 집합은 이미 생성된 객체들의 집합이고, 생성개념에 의한 집합은 앞으로 생성될 객체들 모두를 대상으로 하는 집합입니다.

2.8 객체는 생성 관점과 사용 관점 둘 다를 가지고 이해되고 개념화되어야 합니다.

객체는 생성개념을 통해 생성되고, 사용개념을 통해 사용됩니다.

2.9 소프트웨어개발자는 도메인에서 생성개념과 사용개념을 구분해야 합니다.

Copyright© 2004-2023 뉴테크프라이م(www.umlcert.com), 김현남 (kimhn@umlcert.com) All rights reserved.

도메인 전문가는 대부분의 도메인 객체들을 생성하지 않고 사용만하기 때문에 생성개념과 사용개념을 구분해 주지 않습니다.

하지만, 소프트웨어 개발자는 소프트웨어 객체를 생성하고 사용해야하기 때문에 도메인에서 생성개념과 사용개념을 구분해야 합니다.

모델링의 많은 문제들이 생성개념과 사용개념을 구분하지 못해서 생깁니다.

행위형식화는 생성과 사용에 대한 구분을 핵심적인 근간으로 삼고 있습니다. 또한 '생성사용사고'를 통해 사용된 객체에서 객체의 생성을 볼 수 있도록 합니다.

3절. 생성개념과 사용개념

3.1 클래스는 객체의 생성개념입니다.

객체가 어떤 모습으로 존재할지는 클래스로 정의됩니다. 클래스에 정의된 대로 객체가 생성됩니다.

3.2 연관은 객체의 사용개념입니다.

생성된 객체는 협력을 위해 사용됩니다. 객체는 협력을 위해 링크로 연결되고 협력 시에 어떤 역할을 해야 할지가 링크 역할로 작성됩니다.

링크는 객체의 사용을 위해 요구되는 것이고, 링크의 개념은 연관입니다.

객체들이 어떻게 사용될 지는 연관으로 정의됩니다. 연관에 정의된 대로 객체가 사용됩니다.

연관의 설명은 협력에 대한 부분이 설명되어야 완성됩니다. 협력은 12장, 13장 컬레보레이션에서 설명됩니다.

3.3 객체 다이어그램은 어떤 객체가 존재하고 어떻게 협력을 위해 사용되는지에 대한 팩트를 표현한 것입니다.

3.4 클래스 다이어그램은 객체의 존재와 협력을 위한 생성개념과 사용개념을 표현한 것입니다.

마치기 전에

이번 장에서는 실체에 대응하는 개념들에 대해서 설명했습니다. 또한 생성개념과 사용개념에 대해 매우 높은 강도로 설명했습니다.

사람은 추상적인 것보다 구체적인 것을 잘 이해한다고 합니다.

그래서 우리는 클래스, 데이터타입, 연관과 같은 개념이 무엇인지를 정의하려고 하지 않습니다.

그것들에 대한 구체적인 것 - 객체, 값, 링크와 같은 것 - 들을 먼저 정의하고, 개념들을 대응시켰습니다.

클래스가 무엇인가를 설명하는 것이 아니라 객체가 무엇인지를 설명하고, ‘실체에 대해 이야기하려면 개념이 있어야 하니까 객체에 대한 개념이 필요하다. 객체에 대한 개념이 클래스야.’라고 하는 것입니다.

여러분도 앞으로 이렇게 이야기할 수 있으면 됩니다.

‘생성과 사용’은 모델링의 성공과 실패를 좌우하는 매우 중요한 개념입니다.

‘생성과 사용’이 ‘시간과 공간’만큼 멋지게 다가올 때까지 “객체는 생성되고 사용된다.”를 반복적으로 외쳐보시길 권합니다.