

2장. 실체(Instance) I

(2023)

시작하기 전에

실체는 세 부분으로 나누어 설명됩니다. 이번 장은 실체의 첫 번째 부분으로 **실체에 대해 아는 것을 목표로 합니다. 특히 객체에 대해서 제대로 알아야 합니다.**

좀 더 구체적으로는 다음과 같은 것들을 할 수 있어야 합니다.

1. 객체와 값의 비교 설명합니다.
2. 링크에 대해 설명합니다.

두 번째 부분에서는 링크에 대해서 좀 더 깊이 있게 다루고 실체의 UML 표현들을 다룹니다.

세 번째 부분에서는 전체와 부분에 대해서 다루고, 실체의 좀 더 복잡한 내용들에 대해서 다룹니다.

이 장에서 사용되는 용어들은 다음과 같이 표기됩니다.

우리는 object와 value를 외국어로 간주하고,

object를 객체로,

value를 값으로 표기합니다.

우리는 link를 외래어로 간주하여 링크로 표기합니다.

우리는 identity, state, behavior, property를 외국어로 간주하고,

identity는 정체성으로,

state는 상태로,

behavior는 행위로,

property는 특성으로 표기합니다.

1절. 실체

1.1 모든 존재하는 것들은 목적을 갖습니다.

존재를 특정 철학에 비추어 설명하면, 우연적이거나 무의미한 존재도 있을 수 있습니다. 그러나 소프트웨어 개발자로서 우리는 이런 실체를 다루지 않습니다.

우리가 다루어야 할 실체는 소프트웨어 시스템 내에서 목적을 갖고 존재해야 합니다. 아무 이유 없이 존재한다면 그것은 버그이고, 제거되어야 합니다.

의미 있는 존재라면 아무 이유 없이, 아무 것도 하지 않도록 만들어 지지 않습니다. 분명한 존재 목적을 가져야 합니다.

1.2 존재목적을 갖고 존재하는 것을 실체(instance)라 합니다.

1.3 존재목적을 달성하려면 뭔가를 해야 합니다.

존재목적은 저절로 달성되지 않습니다. 실체가 무엇인가를 해야 합니다.

1.4 실체가 존재목적을 달성하기 위해 해야 하는 일을 행위(behavior)라고 합니다.

1.5 실체는 그 자체로 독립적으로 인식되는 것과 독립적으로 인식되는 것에 의존해서 인식되는 것으로 구분됩니다.

친구인 두 사람이 길을 가면서 대화를 하고 있습니다.

“색깔이 참 노랑기도 하네!”

“무슨 색깔?”

“해바라기의 꽃잎 색깔”



그림 2.1

그림 2.1과 같이 해바라기가 있습니다. 해바라기의 꽃잎 색깔인 노랑도 있습니다. 해바라기도 노랑도 있다는 점에서는 같습니다. 즉, 다 존재하는 겁니다.

근데 이 둘을 따로 떼어 놓고 보면 큰 차이가 있습니다.

해바라기는 그 자체적으로 이 해바라기 저 해바라기 할 수 있습니다. 하지만 해바라기의 색깔에 대해서 이야기 할 때는 먼저 어떤 해바라기인지를 가리켜야 합니다.

1.6 해바라기와 같이 그 자체로 독립적으로 인식되는 실체를 객체(object)라고 하고, 노랑과 같이 독립적 실체에 의존해서만 인식되는 실체를 값(value)이라 합니다.

1.7 객체들은 다른 객체들과 연결되어 있습니다.

연결은 실체와 실체를 연결해야 함으로 실체입니다. 연결하는 것이 목적인 실체입니다.

1.8 객체와 객체사이의 연결실체를 링크(link)라고 합니다.

연결하려면 연결하는 대상을 독립적으로 인식해야 합니다. 연결은 독립적으로 인식할 수 있는 객체들 사이에서만 가능합니다.

1.9 인식은 주어진 환경 즉 컨텍스트(context)에 따라 달라집니다.

컨텍스트에서 그 자체로 독립적으로 인식할 만한 가치가 있기 때문에, 독립적으로 인식할 수 있기 때문에 독립적으로 인식되는 것이고, 그럴 필요가 없거나 그렇게 할 수 있는 방법이 없을 때 의존적으로 인식되는 것입니다.

동일한 주소라고 하더라도

배송회사에서 고객의 주소로 사용될 때와

우체국에서 관리되는 주소로 사용될 때는 인식이 달라집니다.

배송회사에서 주소는 고객에 대한 인식 없이는 인식할 만한 가치가 없지만, 우체국에서 관리되는 주소는 독립적으로 인식할 만한 가치가 있습니다.

1.10 대상이 다루어지는 인식의 컨텍스트를 도메인(domain)이라고 합니다.

객체인지 값인지 구분은 인식의 문제로, 이전에 대상을 어떻게 구분했는지에 대한 경험적 지식보다 도메인이 중요합니다.

1.11 객체와 값 구분은 복잡성을 다루기 위한 것입니다.

대상의 수가 많으면 복잡해 보입니다. 복잡한 것은 한 번에 다룰 수 없습니다.

객체모델링은 대상들 중에 값들을 의도적으로 감춤으로서(무시함으로써) 한 번에 다루어야 할 대상의 수를 줄입니다. 실체들 중에 값들이 객체들 보다 월등히 많기 때문에 이 방법은 매우 효율적입니다.

2절. 객체

2.1 객체는 그 자체로 인식될 만한 가치가 있는 독립적 실체입니다.

2.2 객체는 실체이기 때문에 행위를 갖습니다.

모든 실체를 존재목적에 갖고 그 존재목적은 행위를 통해 달성됩니다.

2.3 그 자체로 존재가 인식되려면, 그것이 그것인 것을 알 수 있어야 합니다.

독립적으로 인식된다는 것은

어떠한 상황에서 그 자체를 지칭할 수 있다는 것입니다.

‘이것’, ‘저것’과 같이 지칭했을 때 그것이 무엇인지를 알 수 있는 방법이 있다는 것입니다.

2.4 그것이 그것인 것을 아는 방법을 자기동일성 또는 정체성(identity)이라고 합니다.

2.5 객체는 정체성을 갖습니다.

2.6 정체성을 갖는 객체는 여러 개 중에서 선택될 수 있습니다.

두 친구의 해바라기에 대한 이야기를 좀 더 들어봅시다.

“색깔이 참 노랗기도 하네!”

“무슨 색깔?”

“해바라기의 꽃잎 색깔”

“어떤 해바라기”

“저기 저 키 큰 해바라기”

“아, 저기 저 키 큰 해바라기”

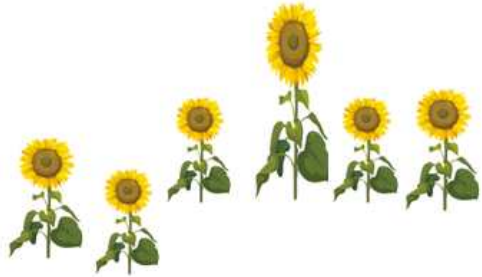


그림 2.2

두 친구는 그림 2.2의 여러 해바라기들 중에서 한 해바라기를 선택했고, 그 해바라기를 지칭하면서 이야기 하고 있습니다.

2.7 정체성을 갖는 객체만 여러 개 중에서 하나를 선택할 수 있습니다.

선택하려면 먼저 대상이 여러 개 있어야 합니다. 여러 대상들 중에 개개의 것을 구분할 수 없다면 선택의 의미가 없습니다.

행위형식화는 객체사고를 통해 모델링 대상들을 정체성이 있는 것(객체)과 정체성이 없는 것(값)으로 구분하도록 하고, 같은 개념으로 불리는 다수의 실체들이 존재하고 그 중에 하나를 선택할 수 있다면 객체로 식별할 수 있도록 합니다.

2.8 변화는 정체성이 전제되어야 합니다.

다음날도 친구인 두 사람이 같은 장소를 지나갑니다.

한 친구가 이렇게 이야기 합니다.

“저 해바라기 꽃잎 색깔이 어제 보다 더 노래졌네.”

어제 본 해바라기의 꽃잎 색깔이 오늘 더 노래졌다고 하려면, 어제

보았던 해바라기와 오늘 본 해바라기가 같은 것이어야 합니다.

오랜만에 친구를 만나서 “야! 많이 변했네!”라고 하려면 지금 내가 말하고 있는 그 사람이 어린 시절의 내 친구인지를 알아야합니다.

2.9 객체에 대해서만 변화를 이야기할 수 있습니다.

정체성을 갖는 실체가 객체이고, 정체성이 있어야만 변화를 이야기할 수 있으므로 객체에 대해서만 변화를 이야기할 수 있습니다.

값은 변하더라도 그 값이 그 값인지를 알 수 없기 때문에, 값 자체를 놓고 변화를 이야기 할 수 없습니다. 값은 객체에 붙어서만 그 객체의 변화를 이야기 하는 데 사용됩니다. 객체에 붙어서 객체의 변화를 이야기할 때도 값은 변화 된 것이 아니라 이전 값이 새로운 값으로 대체된 것 입니다.

2.10 변화 또한 인식되는 것입니다. 인식할 만한 가치가 있는 것만 변화로 인식합니다.

2.11 비교를 통해서 변화를 인식할 수 있습니다.

해바라기의 꽃잎 색깔이 변했다고 하려면 어제의 꽃잎 색깔이 어떠했는지, 오늘의 꽃잎 색깔이 어떠한지를 비교할 수 있어야 합니다.

오랜 만에 만난 친구가 변했다고 하려면 변하기 전의 모습은 어떠했는지, 변한 후의 모습은 어떠한지를 비교할 수 있어야 합니다.

2.12 비교하려면 비교할 수 있는 이름과 값이 있어야 합니다.

해바라기에 대해서 이야기 하던 두 친구 중 한 명이 다음날 다시 그 해바라기를 보고 혼잣말을 합니다.

“어제 보다 더 노래졌네!”

어제의 해바라기 꽃잎 색깔이 노랑이었다면 오늘의 해바라기 색깔은 더 노랑입니다.

해바라기의 변화를 이야기하기 위해 비교 대상의 이름인 ‘꽃잎 색깔’ 과 비교를 위한 ‘노랑’와 ‘더 노랑’ 이라는 값이 있어야 합니다.

객체의 변화를 알려면 비교할 수 있는 값이 필요합니다. 값의 존재 하는 중요한 이유 중 하나입니다. 값이 객체에 의존해서 인식되는 이유가 되 기도 합니다.

2.13 객체의 변화를 이야기하기 위해 사용되는 비교대상의 이름과 값을 객체의 특성(property)이라 합니다.

2.14 변화는 객체 자체를 대상으로 합니다.

해바라기 꽃잎 색깔 변화에 대해서 이야기하고 있지만 결국은 해바라기의 변화에 대해서 이야기하는 것입니다.

해바라기에 어떤 변화가 일어났는데 그 변화는 꽃잎 색깔이 더 노래진 거야라고 이야기하는 것입니다.

2.15 변화를 겪는 객체가 그때 어떠했는지 또는 지금 어떠한지에 대한 표현을 상태(state)라고 합니다.

2.16 객체의 상태는 특성들의 조합으로 표현됩니다.

해바라기의 상태는 꽃잎 색과 키로 표현할 수 있습니다.

2.17 하나의 특성은 여러 개의 값을 가질 수 있습니다.

좋아하는 색깔은 여럿일 수 있습니다.

2.18 다수의 특성을 하나로 묶어 다룸으로 인식의 효율성을 높일 수 있습니다.

배송을 위해 고객의 우편번호, 기본주소, 상세주소를 별개로 다룰 수도 있지만 주소라는 그룹으로 묶어 다룰 수도 있습니다.

주소라는 그룹을 사용함으로 세부적인 항목들에 대한 관심은 감추고 주소만을 인식함으로 인식의 효율성을 높일 수 있습니다.

우편번호, 기본주소, 상세주소가 하나로 묶여 그룹을 이루었지만 이것들은 여전히 고객이라는 객체의 인식에 의존하는 값 입니다.

2.19 하나로 묶인 특성들의 그룹을 데이터값(data value)이라 합니다.

3절. 값

3.1 값은 독립적 실체에 의존해서 인식될 때만 인식할 가치가 있는 의존적 실체입니다.

3.2 값은 실체이기 때문에 행위를 갖습니다.

3.3 값은 정체성을 갖지 않습니다. 정체성을 갖지 않기 때문에 상태 또한 갖지 않습니다.

4절. 링크

4.1 링크는 객체와 객체를 연결하는 연결실체입니다.

4.2 객체와 객체를 연결시키는 것은 나중에 필요할 때 연결시켰던 바로 그 객체를 찾기 위해서입니다.

4.3 나중에 필요할 때 찾으려면 연결이 어느 정도 지속되어야 합니다. 연결이 일시적이라면 연결이 유지되는 그 시점에서만 연결된 객체를 찾을 수 있습니다. 나중에 필요할 때 연결된 객체를 찾아야 한다면 연결이 필요하지 않을 때까지는 연결을 유지하고 있어야 합니다.

4.4 링크는 객체와 객체를 연결하는 강한연결입니다.

우리는 일시적인 연결을 약한연결, 연결이 필요하지 않게 되어 끊어야 할 시점이 되기 전까지 연결을 유지하는 것을 강한연결이라고 합니다.

마치기 전에

이번 장에서는 실체의 세 가지 종류인 객체와 값과 링크에 대해서 설명했습니다.

객체에는 좀 더 많은 분량을 할당했고, 값과 링크는 간략하게 다루었습니다.

객체와 값은 상대적인 것으로 객체를 잘 알면 값도 잘 알 수 있습니다. 링크는 이후의 장에서 좀 더 자세히 다룹니다. 이 장에서는 링크가 이런 거구나 정도만 알고 넘어가면 됩니다.

객체와 값의 구분은 객체모델링에 있어 복잡성을 다루는 가장 기본적인 것으로 객체모델링의 관문이며 첫 단추입니다.

이 문을 통과하는 것 외에 다른 문으로 들어갈 방법은 없습니다.

첫 단추를 잘 꿰어야 나머지 단추들도 잘 쥘 수 있습니다.

이번 장에는 다수의 용어가 등장합니다. 새로운 용어가 나오면 먼저 그 이름과 정의를 외우십시오.

용어정의를 외워야 합니다. 외워야 그 다음을 기약할 수 있습니다.

이번 장은 각 절의 항목 하나하나가 심도 있는 내용들로 채워져 있어 내용을 소화해내기가 쉽지 않을 수 있습니다. 심도 있는 내용을 내 것으로 만드는 가장 좋은 방법은 반복입니다.

반복해서 읽고 읽은 내용을 다른 사람에게 설명해 보는 것입니다. 만족할만한 수준이 될 때까지 반복합니다.

그냥 대충 넘어갈 내용이 아닙니다. 본문의 내용을 세 번 이상 정독합니다.