

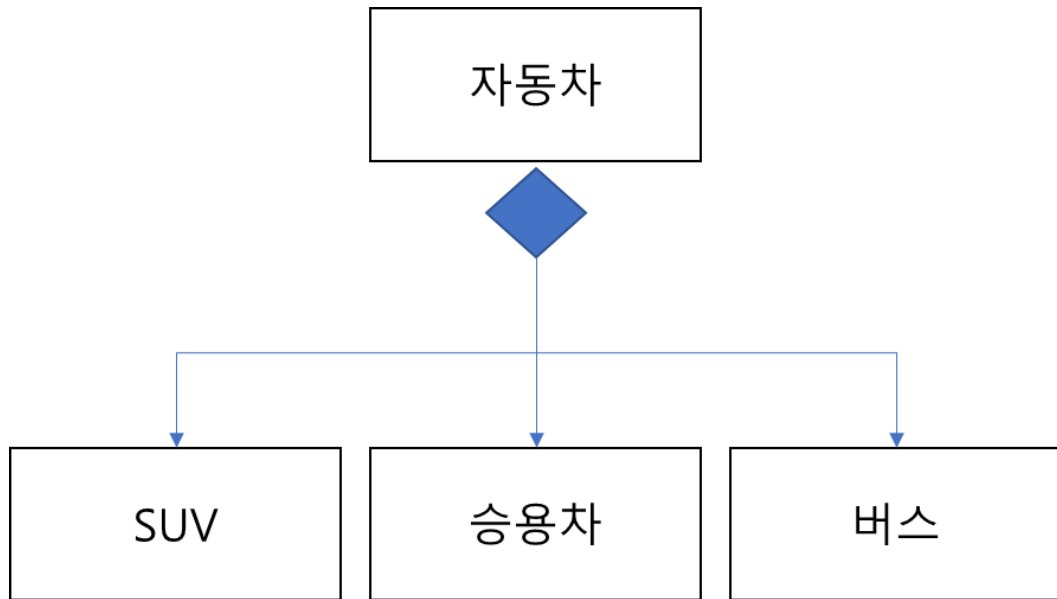
과목 | 객체지향적사고와프로그래밍 학번 | 60205057 이름 | 윤희슬

목차

1. Specialization

2. Aggregation Hierarchy

1. ¹Specialization



종류라는 것은 엔티티들을 일반화 시키고, 일반화시킨 것에서 특화 시킨 것이다. 일반화는 여러 개의 엔티티 중에서 공통성을 뽑는 것을 의미한다. 일반화와 반대의 의미가 특화다.

자바에서는 상속, 확장, 특화가 모두 같은 의미로 쓰이고 있는데, 이는 모두 공통적인 부분을 가지고 있고 거기서 무언가 추가되었다는 의미다. 즉, 속성을 더 가지고 있는 것을 의미한다. 예를 들면 황인종은 인종 중에 노란 피부를 가진 인종이다.

위의 그림에서 자동차 자체가 이미 3 개의 자동차의 공통 부분이라는 의미다 SUV, 승용차, 버스는 각각 자동차에 특성을 가지고 상속, 확장, 특화 시킨 것이다.

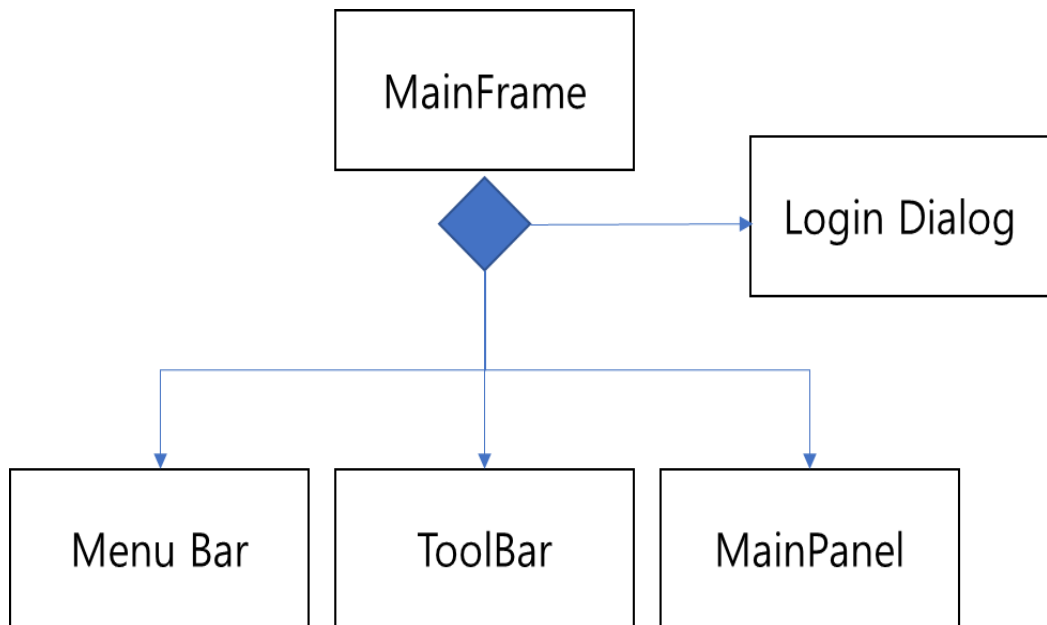
또한 특화는 트리 구조, 계층 구조를 가진다. 따라서 확장하고 초기화하고 자식들을 붙여주는 식의 구조를 가진다.

특화는 결국 생각을 재사용 하는 것이다. 항상 차이를 이용하여 기억하는데, 이런 구조가 없으면 처음부터 모든 것을 다 기억해야 한다. 따라서 이러한 구조를 사용하여 공통성과 체계를 가지면 아주 최소한의 기억을 가지고 일을 할 수 있다. 이러한 장점 때문에 프로그램에서도

¹ <https://velog.io/@yeezze/3주차-1>

소스코드의 재사용에 유용하다. 소스코드의 공통 부분을 그대로 쓰고 그걸 확장해서 사용할 수 있기 때문이다.

2. Aggregation Hierarchy



사용자와 상호작용을 하려면 프로그램이 시각, 청각, 촉각 등의 오감을 통해 정보를 얻어야 한다. 즉, 유저 이벤트를 받아야 한다. 유저 이벤트가 실제로 프로그램에 들어가면 이것이 제이프레임에 전달되고, 제임프레임이 본인의 자식들에게 이벤트를 전달해준다.

이와 같이 유저 이벤트를 처리하기 위해서는 aggregation hierarchy 구조로 이뤄져 있어야 한다.

aggregation hierarchy 는 포함 구조다. 포함은 예를 들면 자동차는 바퀴와 문과 유리 ..등으로 구성되어 있다는 것과 같다. 종류와는 다른 것이다. 이를 프로그램에 적용하면 제이프레임을 상속받은 메인프레임 안에 여러 패널이 있고, 타이틀, 툴바, 메뉴바 등등이 있는 것과 같다.

포함관계는 이러한 구성 요소들을 모아서 연결하면 전체가 만들어진다.

제이프레임한테 무언가를 그리라고 윈도우가 명령을 내리면 제이프레임은 그리라는 명령을 메인프레임에게 내린다. 제이프레임이 메인프레임에 자식들을 추가해놨는데, 이때 메인프레임에 그리라는 명령이 내려지면 메인프레임의 자식들에게도 그리라는 명령이 내려진다.

Aggregation hierarchy 는 view 가 대표적인 예시다. 윈도우한테 애플리케이션의 최상위의 명령권자에게만 명령을 내리면 된다. 왜냐하면 알아서 자식들에게 명령이 전해지기 때문에 구체적으로 모두에게 명령을 내리지 않아도 된다. 이러한 체계를 만들기 위해 aggregation hierarchy 구조를 사용하는 것이다.