

5957 객체지향 프로그래밍

60222117 이서현

목차

1. 관점
2. 객체지향 패러다임
3. Object association
 - A. Object Association
 - B. Object Aggregation Hierarchy
 - C. Class Inheritance Hierarchy
4. 유동소수점

주제 : 객체지향 패러다임

1. 관점

관점이란 세상의 특정 부분만 추출하여 세상을 보는 것을 의미한다. 관점은 복잡한 세상을 단순하게 관찰하기 위해 특정 기준에 따라 세상을 구분하고 정의하는데 사용한다.

예를 들어 기하학에서는 차원을 1차원, 2차원, 3차원으로 나눈다. 차원이란 좌표계로 표현할 수 있는 위상을 의미한다. 위상은 다른 것과 구별되어 고유의 값을 나타내는 것을 의미하며, 이는 좌표를 통해 표현할 수 있다. 1차원, 2차원, 3차원은 몇 개의 좌표계로 위상을 표현하는 지에 따라 구분할 수 있다. 같은 데이터를 1차원과 2차원에 나타낼 때, 1차원에서는 데이터를 일렬로 정렬하는 반면, 2차원에서는 독립변수를 두개 사용해 데이터를 열과 행 형식으로 나타낼 수 있다. 독립변수를 여러 개 사용해 데이터를 나타내면, 원하는 값을 검색할 때, 걸리는 시간을 줄일 수 있다.

2. 객체지향 패러다임

객체지향적 패러다임은 세상을 객체와 객체의 상호작용하는 관점으로 바라보는 것을 의미한다. 다시 말해, 객체지향적 패러다임은 세상은 객체로 구성되어 있어야 하고, 객체가 서로 상호작용한다는 것이다. 따라서 객체지향적 패러다임으로 프로그램을 구성할 때, class를 통해 만든 객체와 이 객체의 상호작용을 실행해야 할 명령어를 통해 statement로 구성해야 한다.

3. Object association

A. Object Association

“Object association은 두 객체 간의 일반적인 관계를 나타낸다.”[참고문헌-1번] 한 객체가 다른 객체의 주소 즉, 포인터를 받아 참조하거나 사용할 수 있는 상태를 의미한다.

B. Object Aggregation Hierarchy

Object Aggregation Hierarchy는 객체 집합 계층 구조를 의미한다. 이는 parent-child 구조로 나타나는데, 하나의 부모 객체의 멤버변수로 자식 객체가 있는 구조를 나타낸다. 부모 객체가 생성되고 소멸될 때를 lifeCycle이라 칭하는데, 객체 집합 계층 구조는 lifeCycle을 공유하지 않는다.

객체 집합 계층 구조는 “has-a”관계를 나타낸다. “has-a”관계는 상위 객체가 삭제되더라도 하위객체는 독립적으로 존재할 수 있는 특성을 가지고 있다.[참고문헌-2번] 이는 객체 집합 계층 구조에서는 lifeCycle에서 부모-자식 순서로 생성되고, 소멸은 자식 – 부모 순서로 소멸되는 composition과는 다르게 한객체가 다른 하나의 객체에 더 크게 종속되지 않기때문이다. 예를 들면 도서관과 책, 주문과 제품 등 하나의 객체가 다른 객체의 구성 요소가 되는 경우이다.

C. Class Inheritance Hierarchy

Class Inheritance Hierarchy는 앞서 이야기한 것과 달리 객체사이 가 아니라 클래스 사이의 관계에서 나타난다. 부모 Class에 있는 구조를 자식 class 구조가 상속받는 것을 의미한다. 객체가 아니기 때문에 속성 값은 상속되지 않지만 구조는 상속된다.

Class Inheritance Hierarchy는 "is -a"관계를 나타낸다. "'Is-a"관계는 상속을 통해 자식 class가 부모 class를 상속하여 자식 class가 부모 class를 확장한다.[참고문헌-3번] 예를 들면, 도형과 사각형, 차량과 자동차와 같이 같은 개념을 상속받는 것을 의미한다.

4. 유동소수점

"유동 소수점은 컴퓨터에서 실수를 표현하는 방법 중 하나이다."[참고문헌-4번] 유동소수점은 소수점 이하를 이진법으로 표현한다. 부호에 1비트, 지수에 8비트, 가수에 소수점 이하를 저장한다. 유동 소수점은 32비트와 64비트의 두가지 형식으로 나누는데 32비트로 표시하는 것이 float, 64비트로 표시하는 것이 double이다. 유동소수점은 소수점 이하를 이진법으로 표현하기 때문에 정확하지 않고 근사값으로 저장한다. 따라서 소수점 이하의 정밀도와 근사성 문제에 주의해야 한다.

참고문헌

- 1)" Object association",chatgpt, <https://chat.openai.com/>, 상단 1번째줄-2번째줄 인용
- 2) "has-a"관계,chatgpt, <https://chat.openai.com/>, 1-3번째줄 인용
- 3) "is-a"관계,chatgpt, <https://chat.openai.com/>, 1-3번째줄 인용
- 4) "유동소수점",chatgpt, <https://chat.openai.com/>, 1번째줄 인용