

# 5957 객체지향 프로그래밍

60222117 이서현

## 목차

1. Object Instantiation
  - A. Object Name Declaration
  - B. Object Body Allocation
  - C. Object Name and Body Binding
2. Polymorphism

주제 : Polymorphism

## 1. Object Instantiation

"Object Instantiation 객체 지향 프로그래밍(OOP)에서 클래스의 인스턴스(또는 객체)를 생성하는 개념이다."[인용- 참고문헌 1번] Object Instantiation에는 name declaration, body allocation, name and body binding이 있다.

### A. Object Name Declaration

Object Name Declaration은 객체를 생성하고, 생성한 객체의 주소를 담을 저장공간의 이름을 정하는 것이다. 객체의 메모리가 할당되고, 그 객체의 메모리에 값이 저장되고, 그 객체를 부를 이름을 정의하면, 앞으로 그 객체의 메모리 안에 있는 저장공간을 간단하게 이름을 통해 부를 수 있다.

### B. Object Body Allocation

Object Body Allocation은 new를 통해 객체의 정보를 저장할 저장공간을 할당하는 것을 의미한다. 객체의 멤버변수, 메서드 등의 정보를 담을 저장공간을 미리 할당할 때 new라는 명령어를 사용하여 앞서 정의한 객체의 이름을 붙여 객체의 body를 부른다. 이렇게 할당한 메모리와 객체의 이름을 연결하는 것을 바인딩이라고 한다.

### C. Object Name & Body Binding

앞서 정의한 객체의 name과 할당한 body의 메모리공간을 연결해서 body의 메모리 공간을 객체의 name으로 부르는 것을 binding이라고 한다. binding에는 크게 Strongly-Typed Language를 사용한 정적 바인딩, Weakly-Typed Language를 사용한 동적 바인딩으로 나눈다.

정적 바인딩이란, name declaration 과정에서 객체의 이름으로 사용할 이름 앞에 Class의 명을 붙이는 것을 의미한다. 이를 Strongly-Typed Language를 사용했다고 하는데, 이는 new를 통해 body 메모리를 할당할 때, 앞에 붙여준 특정 Class의 구조를 따라 메모리를 할당해야 하기 때문에, Strong-typed를 사용했다고 한다. 또한, 앞서 할당할 body의 메모리의 구조를 확실하게 정했고, 변하지 않기 때문에 정적 바인딩이라고 한다.

정적 바인딩은 컴파일 시간에 바인딩이 이루어진다. 즉, 컴파일 시간에 객체의 타입을 확정할 수 있고, 프로그램이 실행되어도 이미 컴파일 시간에 타입을 확정했기 때문에 변화하지 않는다. 또한, 오버로딩을 통해 같은 이름의 메서드를 매개변수의 타입과 개수를 다르게 정의해 더 다양한 메서드를 구성할 수 있다. 이는 하나의 Class 구조로만 할당하기 때문에, 하나의 Class 안에서 메서드를 비교하므로 가능한 것이다.

동적 바인딩이란, 정적 바인딩과 달리 name declaration 과정에서 Class 명을 붙이지 않는 것을 의미한다. 따라서 정적 바인딩과 달리 body의 메모리 할당할 때 꼭 특정 Class의 구조를 따를 필요가 없어 더 자유로운 편이다. 따라서 이를 Weakly-Typed를 사용했다고 칭하고, 메모리의 구조는 new를 통해 어떤 Class 모양으로 만들지 자유롭게 선택할 수 있어 동적 바인딩이라고 한다.

동적 바인딩은 정적 바인딩과 달리 런타임 즉, 실행시간에 객체의 타입을 확정한다. 늦은 바인딩이라고 부르기도 한다. 동적 바인딩에서는 오버라이딩이 가능하다. 이는 다형성과 관련이 있는데, 상속관계에 있는 클래스 간에 같은 이름을 재정의하여, 부모 Class의 구조를 모두 가져오되, 특정 요소만을 수정할 수 있다는 것이다. 오버로딩은 자손의 Class에 맞게 함수를 수정해야 하기 때문에 특정 Class로 정의를 하면 부모의 Class를 불러올 수 없기 때문이다.

## 2. Polymorphism

Polymorphism이란 다형성을 의미한다. 이는 정적 바인딩을 하는 자바에서도 동적 바인딩을 사용하기 위해서 상속을 이용한다. 비록, 동적 바인딩처럼 Class명을 표시하지 않을 순 없지만, Super Class이름을 적어서 super Class와 자식 class를 상속을 통해 연결해주어 동적 바인딩을 이용한다. 따라서 상속 개념을 통해 동적 바인딩을 이용할 수 있고, 오버라이딩까지 할 수 있어서 결론적으로 다형성을 갖출 수 있게된다.

## 참고문헌

1)" Object Instantiation", <https://chat.openai.com>, 1번째줄-2번째줄 인용