

5957 객체지향 프로그래밍

60222117 이서현

목차

1. Instantiation
2. MVC model
3. Client-server

주제 : Instantiation과 mvc model, client server

1. Instantiation

“Instantiation은 class로부터 객체를 생성하는 것을 의미한다.”[참고 문헌 2번 인용]큰 뼈대고 틀인 class에서 실체인 객체를 만들어낸다. 이때 class 안에 있는 멤버변수들은 생성될 때마다 메모리를 할당한다. 따라서 같은 class를 여러 번 Instantiation하면, 여러 개의 겹치는 멤버변수들이 메모리에 저장되어 있다. 그러나, 함수의 경우는 이와 다르다. class안에 있는 메소드는 class를 Instantiation해도 class의 함수 주소로부터 함수를 호출하는 것이다. 하나의 class를 여러 번 Instantiation하더라도 메소드는 한번만 메모리에 저장되어있고, 이 이후에는 모두 이 저장되어 있는 주소를 호출할 뿐, 메모리에 저장하지는 않는다. Class의 멤버변수처럼 생성 할 때마다 이전에 할당했던 멤버변수의 메모리는 잊고, 다시 생성하는 것을 무상태성이라고 한다. 상태가 유지되지 않는다는 뜻이다. 반면, class의 메소드처럼 여러 번 생성해도 이전에 저장되어 있는 곳으로부터 가져오는 것을 상태성이라고 한다. 이전의 상태를 유지한다는 의미이다.

2. mvc model

우리가 서비스를 제공할 때에는 Model, View, controller로 역할을 나누

어 코드를 정리한다. Model은 database에서 즉, 데이터의 저장소에서 data를 꺼내오고 넣어놓는 데이터 저장소 관리사무소이다. View는 사용자에게 결과를 보여주고, 사용자로부터 데이터를 받아오는 것이다.

Controller는 view를 통해 데이터를 받아와서 이 데이터가 유효한지, 그대로 처리해도 되는지를 판단한다.

Controller는 우리가 나름의 규칙을 만들어 데이터의 유효성을 판단하고 exception을 던지거나 요청한 서비스를 수행하는데, 이 규칙을 프로그램이 알아서 만들어 내고 알아서 판단하는 것이 AI, 딥러닝 이다.

3. Client server

Client와 server 가 다른 context에 있을 때 client에서 server의 함수를 호출하려면, 객체의 주소를 알아야하고, 함수의 주소를 알아야한다. 그러나 client와 server가 다른 context라면, client 혼자, server 혼자만으로는 알 수가 없다. 따라서 객체의 주소와 함수의 주소를 받아 연결하고 함수를 호출해야 한다.

이러한 통신의 방법 중 가장 간단한 것은 네트워크 카드를 이용하는 것이다. 네트워크 카드를 꽂아 전기신호로 데이터를 주고받는 것이다. 또, 소켓을 통해 메모리 주소를 공유할 수 있다. 물리적인 IP주소와 논리적인 port number를 이용하면 소켓을 생성할 수 있고, 이 소켓을 통해 context가 다른 객체에 접근할 수 있다. 그러나, 소켓을 통해 IP주소 port number 외에 객체 정보를 계속 입력하게 되면 소켓의 개수는 무한히 많아지기 때문에, 소켓은 ip주소와 port number를 통해 연결하는 역할만 하고, client에서 원하는 객체 정보는 stub을 통해 전달한다. skeleton에는 객체 정보인 string을 각 객체의 주소로 읽기 위해 mapping table을 사용한다. 이 과정에서 mapping table 대신 객체자체를 전달하고 객체 자체를 전달 받는 통신과, class 자체를 전달해서 class를 통해 객체를 생성해서 사용할 수 있도록 하는 통신이 추가적으로 더 존재한다.

참고문헌

- 1) "Java Reflection 개념 및 사용법", tistory, <https://gyrfalcon.tistory.com/entry/Java-Reflection>
- 2) "instantiation", chatgpt, 인용 : 1번째줄