

60200307 우연정

목차

1. MVC 모델
2. 클라이언트 서버 프로그램
  - 2-1. RMI(Remote Method Invocation)
  - 2-2. Object serialization
  - 2-3. Class Transfer
3. Stateless vs Stateful

## 1. MVC 모델

MVC(Model-View-Controller) 모델은 애플리케이션을 세 부분으로 나누어 관리한다. 모델은 데이터와 비즈니스 로직을, 뷰는 사용자 인터페이스를, 컨트롤러는 사용자의 입력과 모델의 데이터를 연결한다. 이와 비슷한 개념으로 쓰리 티어 아키텍처가 있으며, 이는 프레젠테이션, 애플리케이션, 데이터베이스 3가지 층으로 구성된다.

정보처리 시스템은 데이터의 입력, 처리, 출력을 담당하는 시스템이며, 제어시스템은 외부의 장치를 제어한다. 정보처리 시스템의 역할은 데이터의 저장과 가공이다.

클래스는 객체를 생성할 수 있고, 데이터(속성)과 함수(메소드)를 포함하고 있다. 객체는 클래스의 인스턴스로서 실제 메모리에 할당되며, 각 객체는 독립된 상태와 행동을 갖는다.

## 2. 클라이언트 서버 프로그램

클라이언트-서버 모델에서 클라이언트는 서비스를 요청하는 역할이고, 서버는 서비스를 제공하는 역할을 한다. 네트워크를 통해서 데이터를 주고받을 때 소켓 프로그래밍이 필요하며, 이때 IP 주소와 포트 번호를 사용하여 통신한다.

### 2-1. RMI(Remote Method Invocation)

RMI(Remote Method Invocation)는 자바에서 제공하는 API로, 한 자바 가상 머신(JVM)에서 실행되는 객체가 다른 JVM에서 실행되는 객체의 메소드를 원격으로 호출할 수 있도록 하는 기술이다. 클라이언트가 서버의 메소드를 원격으로 호출할 수 있도록 해주며, 컨텍스트가 다른 컴퓨터끼리 통신할 때 사용된다.

### 2-2. Object serialization

“직렬화(Serialization)는 객체의 상태를 바이트 시퀀스로 저장하는 프로세스를 의미한다. 반대로 역직렬화는 이런 바이트를 객체로 다시 구축하는 프로세스다.”  
[<https://www.itworld.co.kr/news/206354#csidxc1965445077c4abb2ce80fe0fc22994>] 객체 직렬화는 객체의 데이터를 영구적으로 저장하거나, 메모리 상태를 파일로 저장하거나, 객체를 네트워크를 통해 전송할 때 사용된다.

### 2-3. Class Transfer

클래스 전송은 특정 클래스의 정의(바이트코드)를 한 환경에서 다른 환경으로 전송하는 것을 의미한다. 이는 주로 클래스 로더가 수행하는 작업이며, 동적으로 클래스를 로드할 때 사용된다. 이는 RMI나 다른 네트워크 기술을 사용할 때, 원격 환경에 해당 클래스가 존재하지 않을 경우 해당 클래스의 바이트코드를 전송하여 로드하는 데 사용된다. 클래스 전송은 네트워크를 통해 클래스의 바이트코드를 전송하고, 대상 환경에서는 이 바이트코드를 사용하여 클래스를 정의하고 인스턴스화할 수 있다

RMI는 객체 간의 원격 통신을 위한 것이며, 객체 직렬화는 객체의 상태를 저장하고 전송하기 위한 것이다. 클래스 전송은 클래스의 정의를 전송하는 것이다.

RMI는 네트워크를 통해 메소드 호출을 전송하는 반면, 객체 직렬화는 객체의 상태를 “바이트

스트림(byte stream, 여러 개의 바이트가 배열되어 있는 데이터 또는 바이트 단위로 입출력이 가능한 파일)”[컴퓨터인터넷IT용어대사전]으로 변환하여 전송한다. 클래스 전송은 클래스의 바이트 코드 자체를 전송한다.

### 3. Stateless vs Stateful

Stateless 방식에서는 각 요청이 독립적이며 서버에 상태 정보가 유지되지 않는다. 반면, Stateful 방식에서는 클라이언트의 상태가 서버에 유지되어, 이전의 상태가 그 후의 요청에 영향을 줄 수 있다.