

# 5957 객체지향 프로그래밍

60222117 이서현

## 목차

1. Remote method Invocation
2. Inter – process communication

주제 : RMI(remote method invocation)

### 1. Remote method Invocation

Remote는 서로 다른 context를 의미한다. 이렇게 context가 다를 때는 서로의 context 안에 있는 내용을 공유할 수 없기 때문에, 통신이 필요하다. 이에 사용되는 것이 RMI이다. 'Remote method Invocation'[참고문헌 1번 인용]은 서로다른 context에 있는 method를 실행시키는 것을 의미한다. 앞서 배운 socket 프로그래밍에선 통신을 위해 IP주소와, port number, 객체에 대한 정보가 필요했다. 그러나 RMI 통신에서는 객체에 대한 정보를 interface를 통해 공유한다. 메소드 이름, 파라미터 정보 등을 Interface를 통해 공유하고, 이 외에 IP주소 및 portnumber, 객체 이름 등은 text 파일을 이용하거나, Rmi 통신의 레지스트리를 이용해 원격객체를 등록하고 연결하는 등 맵핑테이블을 대체한다.

### 2. Inter-process communication

Inter process communication은 프로세스 간에 데이터를 주고받는 것을 의미한다. 이 개념의 구체적인 방법론으로 RMI 및 Socket 통신을 예시로 들 수 있다. 하나의 context 에서 함수를 호출하고 객체를 생성하려면, 객체의 주소를 알아야 하듯, 여러 context 즉, process에서

communication을 하려면 process의 주소를 알아야 한다.

다른 언어와 달리 java는 os가 달라도 서로의 process를 실행할 수 있다. 왜냐하면, java는 각 os에 맞는 기계어로 compile을 하는 것이 아닌, java에서 만든 고유한 기계어로 compile을 하고, jvm을 통해 os가 java 기계어인 바이트 코드를 해석할 수 있도록 환경을 설정하기 때문이다. 실제로 바이트 코드를 실행하는 시점에서는 JRE가 바이트코드를 JIT 컴파일러를 통해 OS에 맞는 기계어로 변환한다.

## 참고문헌

1)“ rmi 통신에서 rebind”, chatgpt, 인용 : 1번째 줄