

# 5957 객체지향 프로그래밍

60222117 이서현

## 목차

1. Naming
  - A. Source Code Level
  - B. Executable Code Level
  - C. Process Level
2. 통신
3. 시나리오 작성

## 주제 : Naming

1. Naming
  - A. Source Code Level (Code Segment)

Source Code Level은 클래스나 변수, 메서드 등의 이름을 선언하는 단계이다. camelCase, 변수나 메서드 이름은 소문자로 시작, Class명은 대문자로 시작, 상수이름은 모두 대문자로 작성하고 언더스코어로 사이를 구분하는 등의 규칙에 맞게 이름을 선언하고, type을 지정하여 값을 대응시키는 과정을 의미한다. 예를 들면 `int x = 3;` 과 같다.
  - B. Executable Code Level (Data Segment)

Executable Code Level은 컴파일 이후의 과정인데, 컴퓨터에서 실행가능한 명령어나 프로그램 코드를 나타내는 용어이다. 위의 예시를 이용해 설명하자면, `int`는 4byte에 해당되기 때문에, 어떠한 특정 메모리의 주소를 기준으로 잡고, 4byte를 특정 주소에 할당해 원하는 값을 넣는다. Executable Code Level에서는 기준으로 잡는 어떠한

특정 메모리의 주소가 특정 값으로 정해지지 않기 때문에 상대주소라고 표현한다. 또한 함수를 호출해서 값을 넣는 것은, jump라는 기계어 명령을 이용하여 함수를 호출하는 것이다.

### C. Process Level (Heap Segment)

앞서 컴파일된 byte code를 실행하면, 내 컴퓨터의 메모리의 0번째 절대주소를 기준으로 잡고, 상대주소들을 모두 나열하여 해당하는 값을 저장하고 이 주소를 변수명에 저장한다. 앞서 상대 주소에 맞게 정렬을 해놓은 메모리들의 나열을, 내 컴퓨터에 있는 0번째 절대 주소를 기준으로 다시 나열한다.

## 2. 통신

결론적으로 "RMI(Remote Method Invocation)"[인용 – 참고문헌 2번] 를 통해 통신하려면, 각각의 메모리주소를 알고 있는 것이 의미가 없고, 공유되고 있는 하나의 고정된 장소에 값을 저장하고 이 주소를 공유하는 것이 의미가 있다. 이는 소켓과 파일을 이용해서 구현할 수 있다. 소켓을 통해서 공간적인 데이터를 공유하고, 파일을 통해 시간적인 데이터를 공유한다. 소켓과 파일을 통해 공유하려면, 이 안에 IP주소와 포트넘버가 있어야한다.

이렇게 IP주소와 포트넘버를 통해 두개의 컴퓨터가 연결되면, objectName, methodName, argument에 대한 이름이 통일되어야 서로 소통이 가능하기 때문에 프로토콜을 만들어야 한다. 이는 Interface를 통해 만들어질 수 있다.

## 3. 시나리오 작성

수강신청 프로그램 시나리오

1. 사용자는 회원가입을 한다.
2. 사용자는 로그인을 한다.

3. 사용자는 강좌를 조회한다.
4. 사용자는 강좌를 선택하여 책가방에 담는다.
5. 사용자는 책가방에 담긴 강좌를 선택해 수강신청 한다.

#### 제약조건

1. 로그인 시 존재하는 account 가 아니면 로그인에 실패한다.
2. 책가방에는 수강 가능 학점 + 6학점만 담을 수 있다.
3. 선 이수 강좌를 수강하지 않은 경우 수강신청 할 수 없다.
4. 수강가능 학점을 초과할 경우 수강신청 할 수 없다.
5. 이미 신청한 강좌와 시간이 겹칠 경우 수강신청 할 수 없다.

#### 참고문헌

- 1) "JVM(Java Virtual Machine), 바이트코드(Byte Code)", tistory, <https://beststar-1.tistory.com/2>
- 2) "RMI의 정의", chargpt, 인용 - 1-2번째 줄