

객체지향 프로그래밍 실습 과제

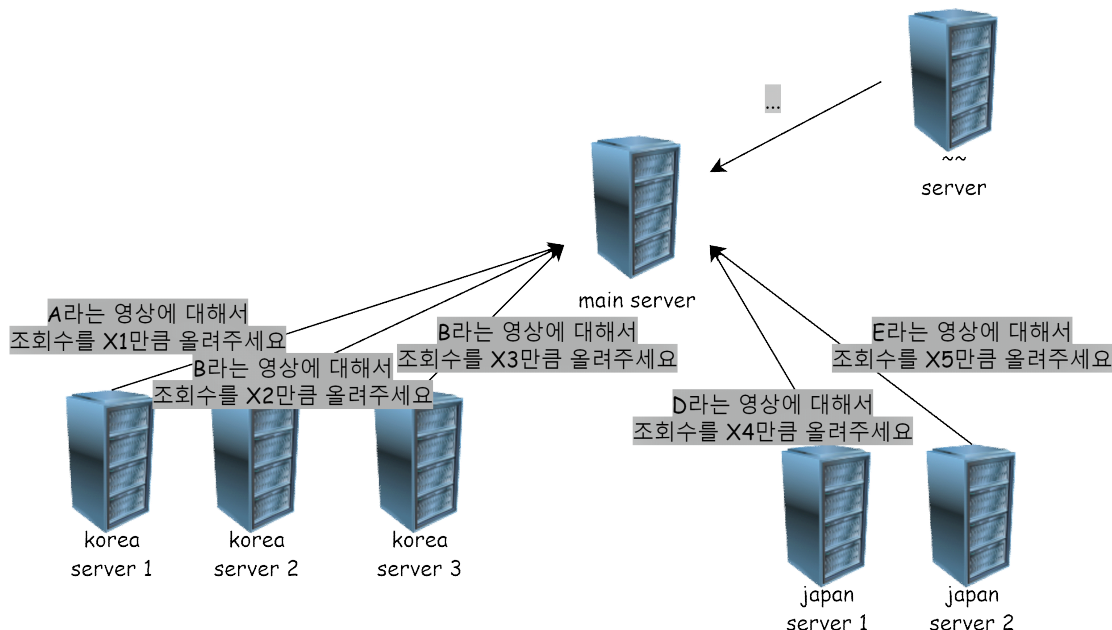
13주차. Concurrency

과제 개요

우리가 구글 본사에 입사하게 되어 유튜브 조회수 관리 프로그램을 개발하게 되었다. 여러 영상이 있을 때, 각각의 영상에 대한 조회수를 정확히 업데이트 할 수 있어야 한다.

우리가 개발할 프로그램은 아래 그림을 따른다. 우리 프로그램은 main server에서 작동할 예정이며, 각 지역에 있는 서버로부터 특정 영상의 조회수를 업데이트해달라는 요청을 받게 된다. 우리는 그 요청에 따라 특정 영상의 조회수를 계속해서 업데이트 시켜나가야 한다.

각 지역 서버의 요청은 ViewUpdateRequest 객체의 형태로 전달되며 이는 ServerConnection 객체를 통해서 받아올 수 있다. 각 지역 서버에서 요청이 단기간에 매우 많이 전달되기 때문에 우리는 멀티 스레딩을 통해서 문제를 해결하고자 한다. 가령 Korea 서버의 요청을 처리하는 스레드 3개, Japan 서버의 요청을 처리하는 스레드 2개 등. 서버의 요청을 처리하는 스레드는 Runnable을 구현하는 ServerProcessor 객체로 생성한다.



과제 세부사항

1. 패키지 준수 사항

1. 제공되는 모든 클래스/인터페이스는 data 패키지에 위치시켜야 한다.
2. ConcurrentRequestProvider는 data 패키지에 위치시켜야 한다.
3. 이외는 자유롭게 위치시킨다.
4. YoutubeData.txt 파일은 이클립스 프로젝트의 루트 폴더에 위치시킨다.

2. public class ViewUpdateRequest – 제공되는 클래스

ViewUpdateRequest
+ VIDEO_NAME: String
+ VIEW_COUNT: int
+ getters

1. ViewUpdateRequest는 서버의 요청사항을 담기 위한 클래스이다.
2. videoName 필드는 요청하고자 하는 영상의 제목을 담는다.
3. viewCount 필드는 해당 영상에 더해져야 하는 조회 수를 담는다.
4. 가령 new ViewUpdateRequest("A", X); 객체는 "A"라는 제목의 영상에 X만큼의 조회 수를 더해야 한다는 요청을 의미한다.

3. public interface RequestProvider – 제공되는 인터페이스

<<Interface>> RequestProvider
+ offer(data: ViewUpdateRequest): void
+ poll(): ViewData

1. RequestProvider 인터페이스는 서버의 요청을 일시적으로 버퍼에 담는 기능을 정의한다.
2. RequestProvider는 ServerConnection 클래스에서 각 지역의 서버에서 들어오는 요청을 담는데 사용된다.
3. offer(ViewUpdateRequest data) 메서드는 서버의 요청을 버퍼에 담는 기능을 수행해야 한다.
4. poll() 메서드는 서버의 요청이 버퍼에 offer 된 순으로 ViewUpdateRequest 객체를 반환해야 한다. 반환된 객체는 버퍼에서 삭제되어야 하며, 더 이상 버퍼에 담겨있는 객체가 없다면 null을 반환해야 한다.

4. public class ServerConnection – 제공되는 클래스

1. ServerConnection 클래스는 각 지역의 서버와 인터넷을 통해 연결되어 ViewUpdateRequest 요청을 받아오는 기능을 수행한다.
2. 메서드의 자세한 기능은 주어진 Javadoc을 참고한다.

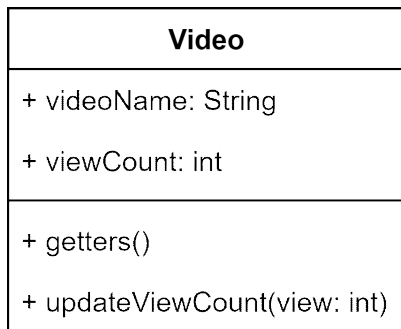
5. public enum ServerType – 제공되는 열거형

1. ServerType enum은 서버 타입을 보다 쉽게 표현하기 위해서 구현되었다.
2. 더 자세한 내용은 첨부된 작년 강의노트를 참고한다.

6. public class ConcurrentRequestProvider

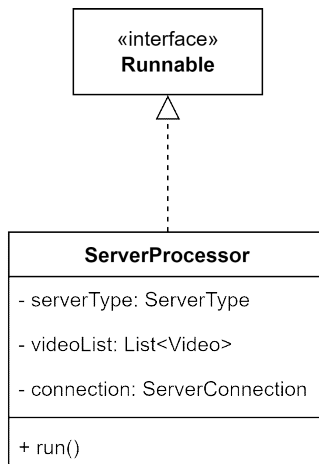
- 1) ConcurrentDataprovider는 RequestProvider 인터페이스를 implement 해야 한다.
- 2) 오버라이딩 해야 하는 메서드의 역할은 2번 문항을 참조한다. 다만, 멀티 스레드 환경에서 동시성을 보장하도록 구현해야 한다.

7. public class Video



1. Video 클래스는 유튜브 동영상의 조회 수 및 여러 속성을 담기위한 클래스이다.
2. 필드로써 아래 사항을 가진다.
 - 1) private String videoName: 영상 제목
 - 2) private int viewCount: 조회수
 - 3) 필요에 따라 추가적인 필드를 사용할 수 있다.
3. 메서드로써 아래 사항을 가진다.
 - 1) public void updateViewCount(int view): viewCount에 view만큼 더한다.
 - 2) 이외 getters
 - 3) 필요에 따라 추가적인 메서드를 구현할 수 있다.

8. public class ServerProcessor



1. ServerProcessor 클래스가 Runnable을 implement 한다.
2. ServerProcessor 클래스는 ServerConnection으로부터 요청을 받아와, 요청사항에 해당하는 영상의 조회수를 업데이트 하는 역할을 수행한다.
3. ServerProcessor 클래스의 serverType 필드는 이 객체가 해당 서버의 요청만을 처리한다는 것을 의미한다. 예를 들어, 어떠한 ServerProcessor 객체가 serverType = ServerType.KOREA 라면, 그 Server 객체는 KOREA 서버의 요청만 처리한다.
4. videoList는 서버에 업로드되어 있는 영상들을 담고 있다.
5. run() 메서드가 아래 사항을 수행하도록 구현한다.

- 1) `ServerConnection.getDataRequest`를 호출하여 `serverType`에 해당하는 서버의 요청사항을 받아온다.
- 2) 요청사항에 알맞게 `videoList`에서 제목이 일치하는 객체를 찾아, 해당 영상의 조회수를 업데이트 한다.
- 3) 위 과정은 `ServerConnection.getDataRequest`가 null을 반환할 때 까지(더이상 반환할 서버의 요청사항이 없을 때 까지) 반복된다.

9. public class Application – Entry point

1. main 메서드를 생성하여 그 안에 코드를 작성한다.
2. 멀티 스레딩은 `ExecutorService`를 사용해서 구현해야 한다.
3. `ServerConnection`을 인스턴스화 한다. 이때, 매개변수로 `ConcurrentRequestProvider.class`를 넣어야 한다. (`ServerConnection` 클래스는 한번만 인스턴스화 될 수 있도록 한다. 여러번 객체생성 해서는 안 됨)
4. `ServerConnection.getExistVideoNames()` 메서드를 사용하여 서버에 업로드 된 모든 영상의 제목을 받아와, 조회수가 0인 `Video` 객체의 리스트를 생성한다. `List<Video>`
5. `ExecutorService`를 사용하여 `ServerType`이 KOREA인 `ServerProcess` 스레드 2개, EUROPE인 스레드 1개, JAPAN인 스레드 2개 NORTH_AMERICA인 스레드 3개를 생성한다. `ServerProcess` 스레드를 생성할 때, `videoList`와 `connection`필드는 각각 4번과 3번에서 생성한 객체를 이용한다. 따라서 모든 `ServerProcess` 객체가 `videoList`와 `connection` 객체를 공유하도록 구성해야 한다.
6. 모든 `ServerProcessor` 스레드가 작업을 끝내고 난 다음에, 모든 `Video`에 대해서 조회수와 제목을 출력한다. (만약 출력되는 내용이 매 실행 시 마다 달라진다면, 동기화에 문제가 있다는 것)
7. 제공된 코드를 제외한 본인이 구현한 클래스에 대해서 동기화 처리를 한다. 이때, 성능에 지장이 없도록 최소한의 부분만 동기화할 수 있도록 한다.

주의 사항

1. 소스코드가 들어있는 Eclipse 프로젝트 폴더와 실행결과 캡처 사진을 압축하여 제출한다.
2. 압축파일 명은 "학번_이름_HW13"으로 한다.
3. Java code convention(Camel case 등)을 준수하고 간단한 주석을 작성한다.

제출 기한

1. 일요일 23:59 까지: 100%
2. 월요일 23:59 까지: 70%
3. 이외: 0%