



응용 소프트웨어전공 60201976 장채은

Client-Server Programming

Index

01

Pipe and Filter

System A 실습

02

Event - Bus

System A 실습
System B 실습

03

Pipe and Filter

Pros and Cons with
클라이언트 서버 프로그래밍과의 비교

04

Event - Bus

Pros and Cons with
클라이언트 서버 프로그래밍과의 비교

05

느낀점

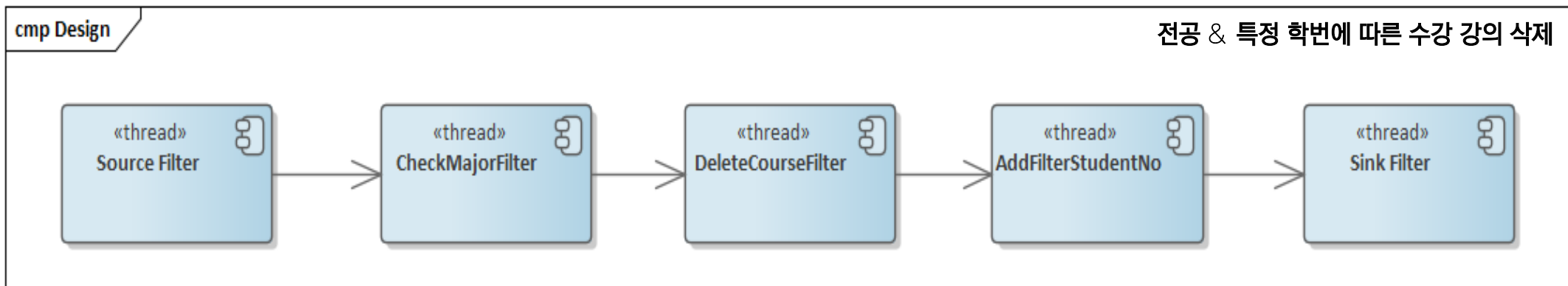
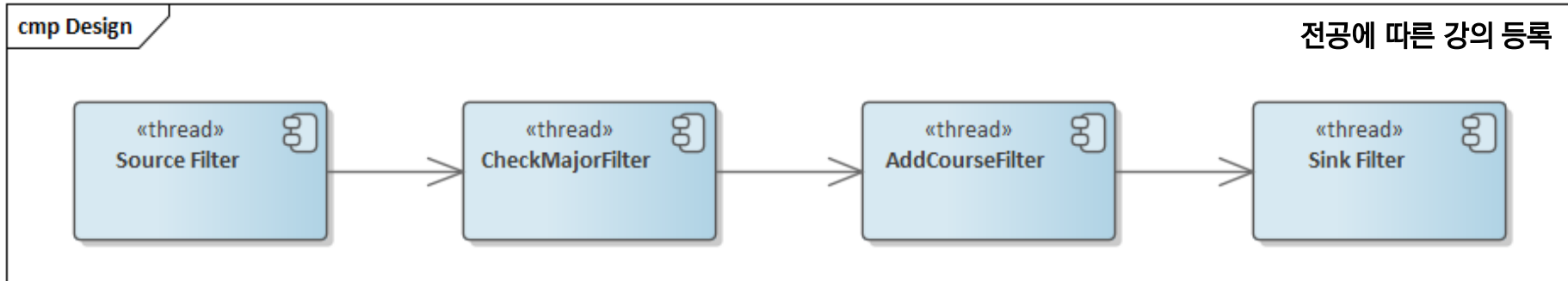
3 개의 Architecture Pattern을
공부하며

01

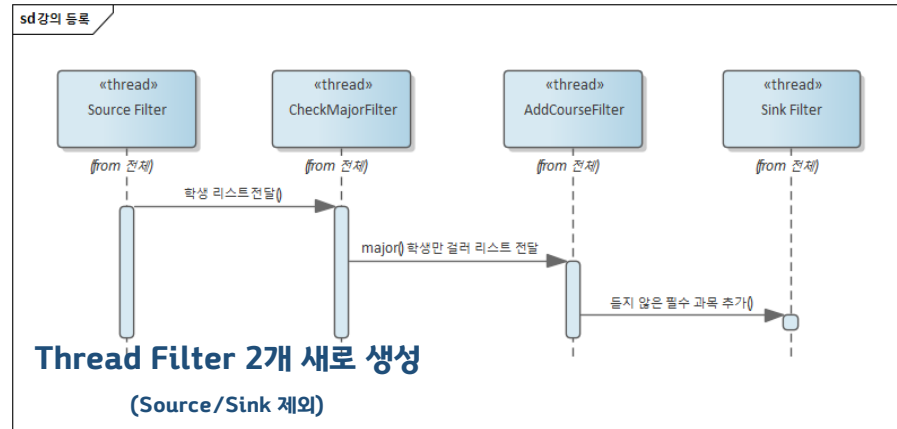
Pipe and Filter

System A 실습

Pipe and Filter 설계 구조



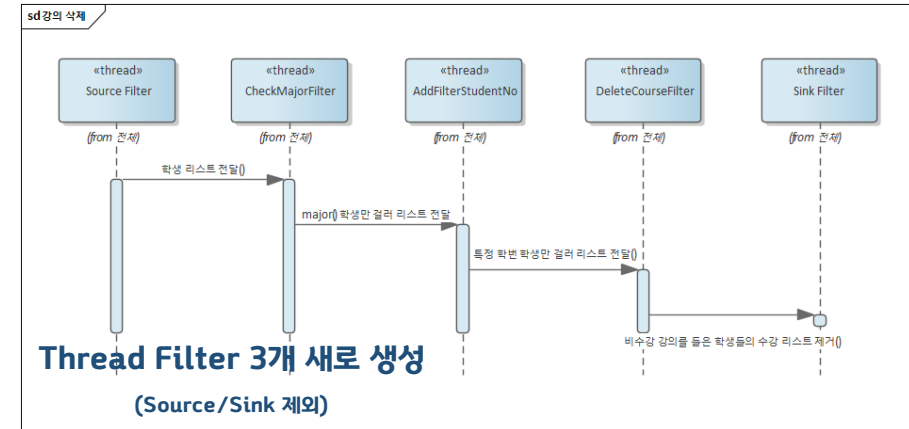
전공에 따른 강의 등록



필수 과목 수강

1. 특정 전공을 전공하는 학생들의 정보
2. 수강한 강의 번호를 확인 후 수강하지 않았다면 새롭게 수강신청

전공 & 특정 학번에 따른 수강 강의 삭제



특정 과목 삭제

1. 특정 전공을 전공하지 않는 학생들의 정보
2. 수강한 강의 번호를 확인
3. 수강했다면 수강신청 철회

전공에 따른 강의 등록

```

20100123 Hwang Myunghan CS 12345 23456 17651 17652
20110512 Park Hongsun CS 12345 23456 17652 17654
20100123 Kim Yunmi CS 12345 17651 17652 17653 23456
20100323 Jung Philsoo CS 12345 23456 17651 17655
20070452 Ko Kyungmin CS 12345 17652 17655 23456
20130091 Kim Chulmin CS 12345 23456 17651 17652
20100128 Kim Minsu CS 12345 17651 17652 17653 23456
20100131 Kim JungMi CS 17651 17652 17653 17655 12345 23456
20130094 Jang Goyoung CS 12345 23456 17653

```

CS 학생 중 12345, 23456을
수강하지 않은 학생 수강한 과목 리스트에 추가

전공에 따른 강의 등록

```

20090421 Kim Jason EE 17651 17652 17653 17654 23456
20080678 Ahn Jonghyuk EE 12345 17651 17652 17654 23456
20120808 Kim Sung Suk EE 12345 23456 17651 17652
20110876 Park Kiyoung EE 12345 17651 17652 17654 2345653 23456
20100131 Kim JungMi CS 17651 17652 17653 17655 12345 23456
20130094 Jang Goyoung CS 12345 23456 17653

```

CS가 아닌 학생 중 23456을
수강하지 않은 학생 수강한 과목 리스트에 추가

전공 & 특정 학번에 따른 수강 강의 삭제

```

20130095 Kim Soyoung EE 12345 23456

```

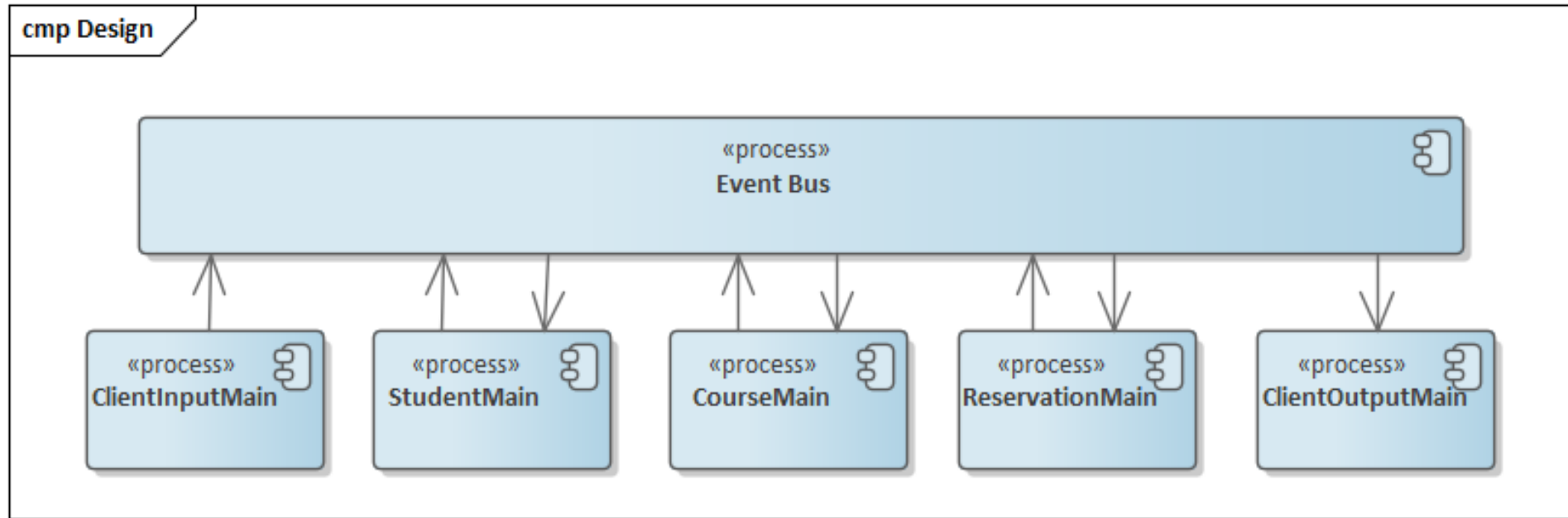
CS가 아닌 13학번 중,
17651이나 17652을 수강한 학생을 찾아 수강한 과목 삭제

02

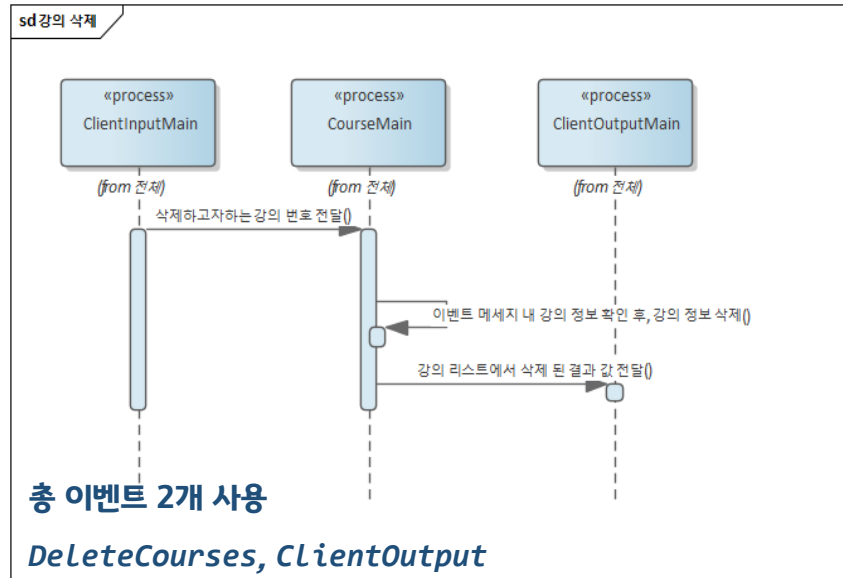
Event - Bus

System A System B 실습

Event – Bus 설계 구조



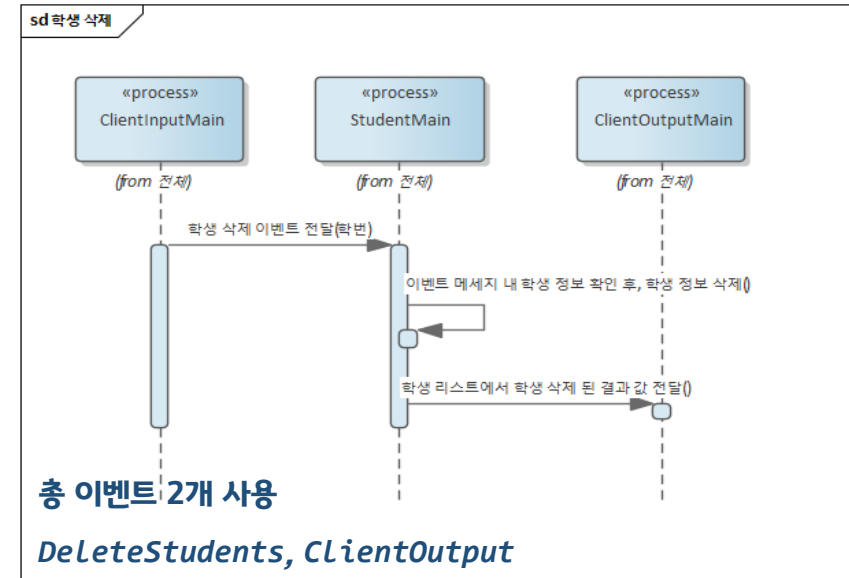
강의번호 전달 → 강의 삭제



삭제하고자 하는 강의 번호 전달

1. 강의 번호를 이벤트에 담아서 전달
2. 이벤트 내 메시지 강의 정보 확인
3. 강의 리스트 내 강의 삭제 후 output 전달

학번 전달 → 학생 삭제



삭제하고자 하는 학번 전달

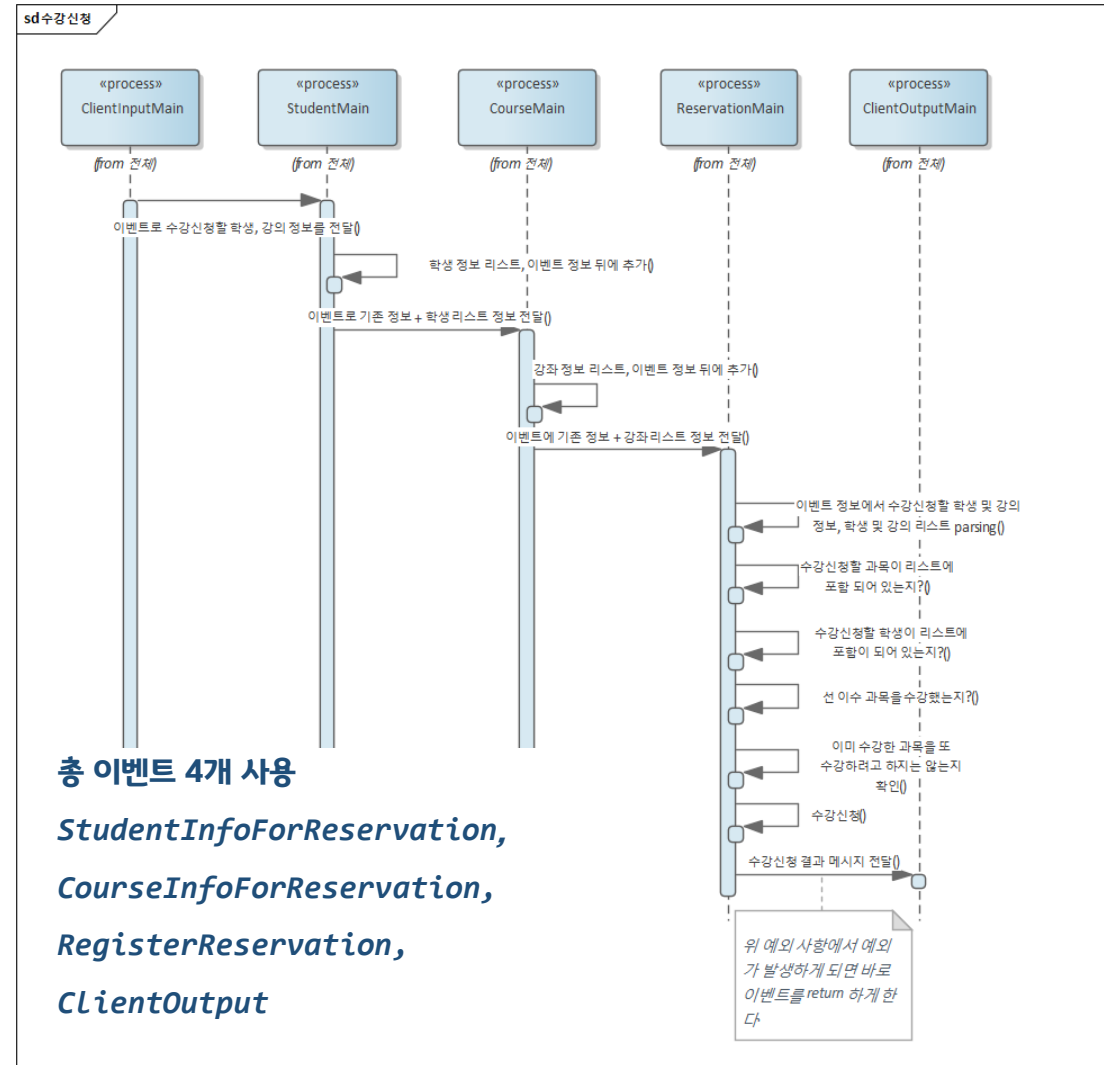
1. 학번을 이벤트에 담아서 전달
2. 이벤트 내 메시지 학번 정보 확인
3. 학생 리스트 내 학생 삭제 후 output 전달

수강 신청

학번 및 수강 희망 강의 번호 전달

1. 학번 및 수강 희망 강의 번호를 이벤트에 담아서 전달
2. 1번의 정보 + 학생 리스트 정보 이벤트 담아서 전달
3. 2번의 정보 + 강의 리스트 정보 이벤트에 담아서 전달
4. 이벤트 Parsing (학번, 수강 희망 강의 번호, 학생 리스트, 강의 리스트)
5. 수강신청 과목이 수강 리스트에 포함 되어 있는지 확인
6. 수강 신청할 학생이 학생 리스트에 포함 되어 있는지 확인
7. 선 이수 과목을 수강했는지 확인
8. 이미 수강한 과목을 또 수강했는지 확인
9. 수강 신청
10. 수강신청 결과를 이벤트에 담아서 전달

System B Enterprise Architect



강의번호 전달 → 강의 삭제

```

** ClientOutputMain (ID:1670815021182) is successfully registered...
// 학생 정보 List
20100123 Hwang Myunghan CS 12345 23456 17651 17652
20090421 Kim Jason EE 17651 17652 17653 17654
20110512 Park Hongsun CS 12345 23456 17652 17654
20100123 Kim Yunmi CS 12345 17651 17652 17653
20100125 Kim Hokyoung ME 12345 23456 17651 17654
20100323 Jung Philsoo CS 12345 23456 17651 17655
20080678 Ahn Jonghyuk EE 12345 17651 17652 17654
20110298 Lee Mijung ME 12345 23456 17651 17652
20120808 Kim Sungsook EE 12345 23456 17651 17652
20080603 Park Kitea ME 17651 17652 17653
20070452 Ko Kyungmin CS 12345 17652 17655
20130091 Kim Chulmin CS 12345 23456 17651 17652
20110876 Park Kiyoung EE 12345 17651 17652 17654
20100128 Kim Minsu CS 12345 17651 17652 17653
20100131 Kim JungMi CS 17651 17652 17653 17655
20130094 Jang Goyoung CS 12345 23456 17653
20130095 Kim Soyoung CS 12345 23456 17651 17652

// 20110876 삭제
This student is successfully deleted.

// 삭제 후 학생 정보 List
20100123 Hwang Myunghan CS 12345 23456 17651 17652
20090421 Kim Jason EE 17651 17652 17653 17654
20110512 Park Hongsun CS 12345 23456 17652 17654
20100123 Kim Yunmi CS 12345 17651 17652 17653
20100125 Kim Hokyoung ME 12345 23456 17651 17654
20100323 Jung Philsoo CS 12345 23456 17651 17655
20080678 Ahn Jonghyuk EE 12345 17651 17652 17654
20110298 Lee Mijung ME 12345 23456 17651 17652
20120808 Kim Sungsook EE 12345 23456 17651 17652
20080603 Park Kitea ME 17651 17652 17653
20070452 Ko Kyungmin CS 12345 17652 17655
20130091 Kim Chulmin CS 12345 23456 17651 17652
20100128 Kim Minsu CS 12345 17651 17652 17653
20100131 Kim JungMi CS 17651 17652 17653 17655
20130094 Jang Goyoung CS 12345 23456 17653
20130095 Kim Soyoung CS 12345 23456 17651 17652

```

학번 전달 → 학생 삭제 & 수강신청

```

** ClientOutputMain (ID:1670815021182) is successfully registered...
// 강의 정보 List
12345 Park Java_Programming
23456 Park C++_Programming
17651 Kim Models_of_Software_Systems 12345
17652 Ko Methods_of_Software_Development 23456
17653 Kim Managing_Software_Development
17654 Ahn Analysis_of_Software_Artifacts 17651
17655 Lee Architectures_of_Software_Systems 12345 17651

// 23456 삭제
This course is successfully deleted.

// 삭제 후 강의 정보 List
12345 Park Java_Programming
17651 Kim Models_of_Software_Systems 12345
17652 Ko Methods_of_Software_Development 23456
17653 Kim Managing_Software_Development
17654 Ahn Analysis_of_Software_Artifacts 17651
17655 Lee Architectures_of_Software_Systems 12345 17651

// 수강 신청 성공
This course is successfully added.

// 수강 신청 리스트
20090421      12345

```

03

Pipe and Filter

Pros and Cons with
클라이언트 서버 프로그래밍과의 비교

**성능이 좋음**

한 Process 내에서 여러 Thread 사용 \leftrightarrow 다수의 Process

Filter 재사용/코드 추가

독립적이기 때문에 한 번만 생성하면 기능이 필요할 때 재사용

다른 시스템에 영향을 주지 않음

Thread가 한 기능 \leftrightarrow 한 소스 파일 내 서로 유기적 연결

기능이 병렬적 진행

같은 시간 다른 기능 진행 가능 \leftrightarrow 한 기능 후 다른 기능 진행

디버깅 가능

\leftrightarrow 디버깅 어려움

**바이트 형태의 데이터 전달하여 복잡**

한 바이트 씩 저장하여 전달 \leftrightarrow 인터페이스 전달

에러 처리 어려움

에러 메시지 전달 어려움 \leftrightarrow Exception 처리 가능

많은 Filter \rightarrow 시스템 부하

정보 결과 값을 Filter에서 변환 (선행 \rightarrow 후행) \leftrightarrow
한 소스 파일에서 모든 작업 가능(ex> Server 내 수강신청 기능)

데이터가 한 쪽으로 흐름

응답, 요청 기능 부적합 \leftrightarrow 응답, 요청 기능 적합

전송시간 느림

04

Event - Bus

Pros and Cons with
클라이언트 서버 프로그래밍과의 비교

PROS

이벤트로 메시지 전달

요청하고 응답하는 프로세스가 서로 다름 \leftrightarrow
인터페이스로 정보 전달 후 같은 프로세스에 응답 받음

유지 보수성/변경이 쉬움

\leftrightarrow 이벤트 버스에 비해 어려움

정보 보낸/받는 사람을 알 필요 없음

이벤트만 체크 \leftrightarrow 정보를 보낸 사람과 받을 사람을 알아야 함.

병렬 전송 유리

흐름이 한쪽으로 이어짐 \leftrightarrow 흐름이 양쪽으로 상호작용

분산화 된 프로그램에서 유리

코드들 사이의 결합도가 낮아 관심사의 분리 가능 \leftrightarrow 결합도가 높음.

CONS

예측 어려움 / 이벤트 낭비 / 복잡함

이벤트 버스에 정보를 모두 보냄 \leftrightarrow 특정 위치, 경우에만 정보 전송

디버깅 안됨

오류 발생했을 때 디버깅 안됨 = 디버깅 안됨

정보 Parsing

이벤트 메시지 Parsing (String 단위) \leftrightarrow 인터페이스로 전달

무거움/메모리 수요 증가

여러 Process를 RMI 통신을 통해 연결 + 하나의 이벤트 모든 노드
= RMI 통신

05

느낀점

3 개의 Architecture Pattern을
공부하며

1. 3개의 패턴의 장단점을 비교하며 필요한 서비스에 따라 어떤 패턴을 사용하는 것이 좋을지 생각해보게 됨.
2. 실습의 경우 클라이언트 서버가 다른 두 패턴을 사용하는 것보다 훨씬 더 효율적일 것이라고 생각함.
3. 이 3개의 패턴 말고도 다른 SW 아키텍처 패턴도 궁금해짐.

Thank you