

### 문제 1. 소감 답변

1달동안 수업에 참여를 한 결과 머리가 멍 했습니다

하지만 어려울수록 이 것들을 나의 것으로 만든다면 더 큰 이득이 된다는 것도 알기에 열심히 버티고 있으나 역시 쉽지는 않았던 것 같습니다

아직 5개월정도 남았고 시작할 땐 엄청난 의지와 새벽6시에 커밋하던 뜨거운 열정으로 시작했지만 높은 역치로 인한 약간의 좌절과 의지박약 패시브로 인해 약간 식었던 거 같습니다 하지만 다시 한번 불태워 보겠습니다

### 문제2. 이론 답변

**DDD**를 이용한다면

끊임 없는 팀원간 회의를 통해 뭘 개발해야는지 어떤 방향으로 나가하는지 정확한 방향을 잡을 수 있고

다른 파트를 개발하고 있는 다른 팀원 등이 볼 때 어떤 부분이 어떻게 개발되고 있는지 빠르게

파악이 가능 따라서 개발 속도 또한 빠르다

개발후에 클라이언트가 추가적인 기능 등을 요구할 시 빠르고 정확하게 업데이트,추가 개발이 가능하다

### 문제3. 이론 답변

**TDD**는 테스트를 먼저 작성하고 그 후에 테스트가 이상이 없으면 그 코드를 사용하는데 테스트를 먼저하기 때문에 결함을 사전에 방지할 수 있다

코드를 수정할 때도 테스트를 하여 잘 작동하는 확인할 수 있다

자기 파트를 먼저 개발을 완료해서 메인이 없어 잘 작동하는 지 확인이 가능하다

테스트는 코드의 예상 동작을 설명하기 때문에 코드에 대한 기능 설명이 적혀있어 다른 사람이

이 코드는 뭘 하는 지 파악할 수 있다

### 문제4. 이론 답변

**DDD** 와 **TDD**는 상호보완작용을 한다 공통적인 목표는 빠르고 정확한 개발하게 해 소프트웨어의 품질을 향상시키기 위해 사용된다

**DDD**에서 생성한 코드 실제 기대한 대로 동작하는지 테스트를 통해 확인할 수 있다

**DDD**로 핵심 도메인을 만들어 **TDD**에서 이 코드의 의미를 더 명확하게 할 수 있다

### 문제5. 고찰 답변

쌍방향 의사소통이 필요하다 누가 ‘해’ 라는 단방향 지시가 아니라 서로 의견을 제시하고 고민,검토를 해 올바른 방향을 잡는 것이 중요한 것 같다

상호 존중

팀원을 존중하지 않는다면 불화가 일어나고 불화가 일어나면 협동을 제대로 할 수 없다

목표 설정 과 역할 분담

지금 무엇을 하는지 무엇을 해야할지 정확한 목표를 설정해야 그 목표를 향해 개발을 진행하고 다른 길로 빠지지 않는다 또 정확한 역할 분담을 해야 아무 것도 하지 않는

사람이나 여러 사람이 같은 파트 개발을 하는 등 시간 낭비를 막고 진행속도를 올릴 수 있다

효율적인 도구 활용

백로그 작성, 프로젝트 개발에 도움이 되는 툴등을 사용해 협업에 도움을 줄 수 있다

또 자신이 맡은 파트를 수행하기 위한 개인 역량 또한 필요한 것 같다

#### 문제6. 이론 답변

어느 부분이 문제가 생겼을 때

함수들의 종속성을 막고 연쇄 작용으로 인해 프로그램 전체가 작동 하지 않는 것을 막기 위함이다

예를 들어 함수 포인터 테이블을 이용해 1번 2번 3번을 골라서 해당 기능을 실행할 수 있다  
1 2 3 기능이 다 잘 작동하고 있었는데 갑자기 3번에서 문제가 발생해 실행되지 않는다고 한다

만약 3번이 다른 함수와 연결돼있었다면 그대로 프로그램은 작동하지 않았을 것이다  
하지만 함수 포인터 테이블을 사용한다면 3번이 안되도 1번 2번은 작동하고 프로그램도 작동한다

#### 문제7 이론 답변

**virtual method**를 통해 메서드를 재정의하여 같은 메서드가 다른 동작을 할 수 있게 할 수 있다

그래서 기존 클래스를 수정하지 않고 기존 클래스의 동작을 확장하거나 변경할 수 있다

#### 문제8 고찰 답변

일단 문제가 뭔지 왜 일어난건지 정확히 파악을 한다

심각한 문제인지 급한 문제인지 주어진 시간이 얼마나 남았는지 등을 파악한다

팀원들과 의사소통을 해 나의 문제를 알려 팀원들의 의견을 들어본다

이 문제를 어떻게 해결할건지 머리를 맞대어 긴급 회의를 해보고

괜찮은 해결방안이 나온다면 빠르게 역할 분담을 하여 문제 해결을 시작한다

하지만 아무리 생각해봐도 해결 방안이 나오지 않는다면 해당 문제에 관한 전문가를 모셔와 협력하거나

외부에 지원을 요청할 것 같다

#### 문제9 고찰 답변

협업을 할 때 백로그를 작성하면 내가 맡은 파트 아닌 다른 사람의 생각을 볼 수 있다

내가 맡은 파트와 다른 사람이 맡은 파트를 보고 저사람이 이렇게 할 예정이나 이런 계획이구나를

파악하여 그에 맞는 기능, 디자인 등등을 개발하여 더 좋은 시너지를 낼 수 있다

또 방향이 현재 가는 방향이 맞는 보면서 계속 체크하고 수정할 수 있으며 다른 사람들이 백로그를 보고 그 사람들의 의견또한 수렴할 수 있다

#### 문제10 고찰 답변

근본적인 이유는

팀원과의 소통부재와 개발 방식인 것 같다

백로그 작성도 일종의 팀원과 소통이라 생각하는게 백로그를 쓰지 않고 각자의 자리에서 정확한 방향과 역할 분담 없이 열심히 코드를 치고 있는데

알고보니 필요 없었던 것이거나 중복 개발을 하고 있었던 등 시간을 낭비하게 되고

어찌저찌 역할 분담을 해서 개발을 하는데 변수 이름이나 코드를 합칠 때 지켜할 것등이 지켜지지 않아 합치지 못하거나 다시 상당부분을 고치거나 어느 부분만 작동하는 등의 결과가 발생했던거 같다

#### 문제11 고찰 답변

기능단위 백로그를 쓰게 된다면 기능에만 집중해 큰 방향을 잃게 되는 것 같다  
같이 작업 중인 다른 팀원들이 백로그를 봐도 왜 이 기능이 필요한지 파악할 수가 없어  
개발 진행 속도 또한 지체될 수 있을 것 같다  
그래서 개발 중에 이 기능이 필요 없었던 것이었다면 시간또한 낭비한 것이 된다

[ 복합 문제 12 ~ 20 ]

<https://github.com/EDDI-RobotAcademy/SDC-Comprehensive-Evaluation/commit/237b31b69d74002a37f0c92ba7e5cd65e564fc1d>

문제 21 고찰 답변

현재 시간과 날짜,요일을 넣어서 데이터를 저장하면 될 것 같다  
특정 시간마다 db에 현재 판정한 사람의 수와 현재 시간과 날짜,요일을 같이 저장한다면  
어느 시간대에 사람들이 제일 많이 몰리는지 무슨 요일에 이 정거장에 사람이 제일 많이  
몰리는지  
계속 db에 저장을 하고 본다면 도움이 될 것 같다

문제 22 이론 답변

자원은 무한하지 않기 때문이다 포인터를 사용하면 유한한 메모리를 효율적으로 사용할 수  
있다  
또 복잡한 데이터구조를 다루는데 유용하다  
함수 포인터를 이용할 수 있고 효율적인 개발 방법인 DDD를 할 때 꼭 필요한 요소이다

문제 23 이론 답변

서로 다른 함수들이 스택을 공유하면 한 함수에서 사용하는 지역 변수 매개변수의 값이  
다른 함수에서 사용하는 값과 충돌할 수 있다 충돌한다면 버그가 생길 수 있다

[ 복합 문제 24 ~ 25 ]

<https://github.com/EDDI-RobotAcademy/SDC-Comprehensive-Evaluation/commit/237b31b69d74002a37f0c92ba7e5cd65e564fc1d>

[ 복합형DdD 문제 26~30 ]

문제33 고찰 답변

도메인을 분리해야될 것 같다...?

[ 복합 문제 34~36 ]