**Software Engineering Q/A Sheet (#7)**

date: 2023/10/30 number: 2018312280 name: 이상수

**Questions from Prof.**

1. Describe the strengths and weaknesses of the object-oriented paradigm.

강점 :

소프트웨어 재사용. 각각의 오브젝트로 나누어진 코드들을 재사용할 수 있다.

유지보수의 측면에서 각 오브젝트 부분들의 수정을 통해 문제가 발생 했을 때 피해를 최소화한다.

Cohesion, Coupling기준의 디자인도 향상시킨다. Cohesion은 시스템을 구성하는 모듈들의 내부적인 연관성, 그리고 coupling은 모듈과 모듈 간에 결합성을 의미한다.

Cohesion은 높을수록, Coupling은 낮을수록 좋다. 이 두가지를 이루기 좋은 방법이 OOP이다.

단점 :

구조 설계에 오랜 시간이 걸릴 수 있다. 각각의 오브젝트 간에 연관성, 결합성등을 잘못 설계하면 문제가 생긴다.

그렇기에 모든 시스템에 사용되지는 않고, 소프트웨어 시스템 중 유지보수가 중요하고 긴 시간 사용될 시스템들 위주로 사용된다.

1. Apply the OMT to your term project just with a small example. (Practice as a team) 스크린샷, 텍스트, 도표, 사각형이(가) 표시된 사진

   자동 생성된 설명

시나리오: 1. 유저가 사이트에 접속 2. 유저가 코드를 입력 3. 코드 결과가 backend에서 도출 4. 결과 출력과 DB에 결과 저장. 5. 이 결과는 유저가 보거나 운영자가 볼 수 있음.

Questions from your ownself

1. OOP 에서 모듈간에 cohesion이나 coupling을 높이고 줄이는 방법은 무엇이 있을 까?

내 답: 모듈간에 무언가 바뀌었을 때 다른 모듈이나 모듈 내에 다른 시스템에 어떤 영향을 끼치는지 분석하고 파악하여 수정한다.