**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)

- Object identification

Problem:

사용자가 자신의 자바코드를 웹사이트에 입력시키면 서버에서는 해당 코드를 직접 실행시킨 후 탄소 배출량을 계산하여 반환한다.

Class Extraction:

- 웹사이트

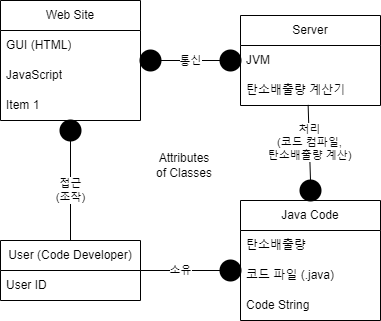
- 서버

- 자바 코드

- 사용자 (코드 개발자)

- Relationships among classes

- Attributes of classes



### Dynamic Modeling

1. Scenario composition

- 사용자는 웹사이트 메인 페이지에 접속하여 자바 코드 입력 페이지에 접속한다.

- 사용자는 자바 코드를 웹사이트에 입력한다

- 웹사이트는 자바 코드를 서버에 제출한다

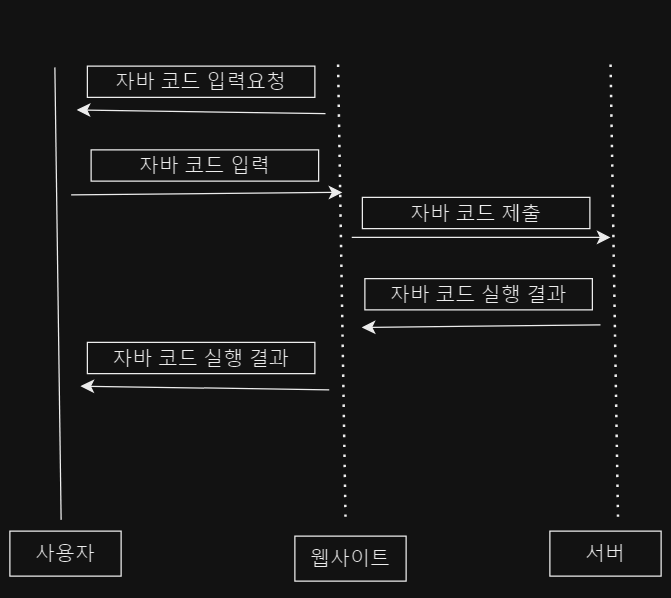
- 서버에서는 자바 코드를 컴파일 및 실행한다

- 서버는 코드가 정상적으로 실행되지 않으면 오류 메시지를 보여준다.

- 실행 성공시, 실행 결과를 바탕으로 탄소배출량을 계산한다

- 결과 값을 웹페이지에 리턴한다

- Sequence Diagram



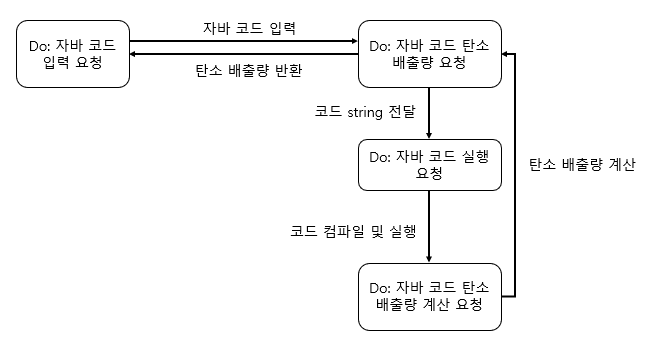
- State Diagram

- 1) do: 자바 코드 입력 요청 → 자바 코드 입력 → 2)

- 2) do: 자바 코드 탄소 배출량 요청 → 코드 string 전달→ 3) & → 탄소 배출량 반환 → 1)

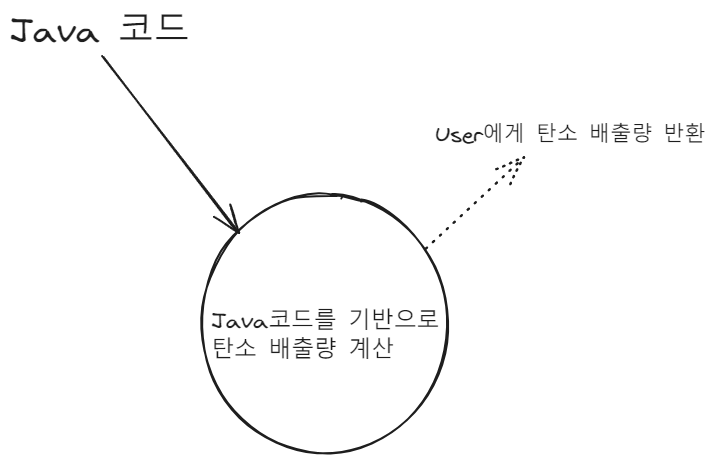
- 3) do: 자바 코드 실행 요청 → 코드 컴파일 및 실행 → 4)

- 4) do: 자바 코드 탄소 배출량 계산 요청 → 탄소 배출량 계산 → 2)



### Functional Modeling

- Data Flow Diagram





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.