

Hero Tech Course

Pre-Course 짭먹

20222191 임채성

우리는 왜 이 과제를 하는가?

- 대부분이 그냥 시켜서 하는 것이겠지만

우리는 왜 이 과제를 하는가?

- ~~대부분이 그냥 시켜서 하는 것이겠지만~~
- 좋은 SW를 만드는 개발자가 되기 위함.

우리는 왜 이 과제를 하는가?

- ~~대부분이 그냥 시켜서 하는 것이겠지만~~
- 좋은 SW를 만드는 개발자가 되기 위함.
 - 그렇다면 무엇이 좋은 SW인가?

좋은 SW 설계란?



좋은 SW 설계란?

- TODO:
본인이 RacingCar 게임 구현에 사용한
클래스 갯수와 이름을 채팅방에 적으세요.



45년 짬밥으로 깨달은 것

2025. 5. 1.

<https://puleugo.dev>



45년 짬밥으로 깨달은 것

1. 고객은 니가 짬 코드 1도 관심없다.



45년 짬밥으로 깨달은 것

1. 고객은 니가 짬 코드 1도 관심없다.
2. 고객은 제멋대로다.



45년 짬밥으로 깨달은 것

1. 고객은 니가 짬 코드 1도 관심없다.
 - 1달 후에 코드 설명하라 했을 때 얼تان다면
2. 고객은 제멋대로다.



45년 짬밥으로 깨달은 것

1. 고객은 니가 짠 코드 1도 관심없다.
 - 1달 후에 코드 설명하라 했을 때 얼تان다면
→ 개발자를 갈구면 해결된다.
2. 고객은 제멋대로다.



45년 짬밥으로 깨달은 것

1. 고객은 니가 짬 코드 1도 관심없다.
 - 1달 후에 코드 설명하라 했을 때 얼탄다면
→ 개발자를 갈구면 해결된다.
2. 고객은 제멋대로다.
 - 요구사항을 제멋대로 바꾼다.



45년 짬밥으로 깨달은 것

1. 고객은 니가 짬 코드 1도 관심없다.
 - 1달 후에 코드 설명하라 했을 때 얼탄다면
→ 개발자를 갈구면 해결된다.
2. 고객은 제멋대로다.
 - 요구사항을 제멋대로 바꾼다.
→ 고객을 갈굴수는 없다.



45년 짬밥으로 깨달은 것

1. 고객은 니가 짤 코드 1도 관심없다.
 - 1달 후에 코드 설명하라 했을 때 얼탄다면
→ 개발자를 갈구면 해결된다.
2. 고객은 제멋대로다.
 - 요구사항을 제멋대로 바꾼다.
→ 고객을 갈굴수는 없다
→ 확장(기능 추가)를 고려해서 작성하자.



그래서 좋은 SW 설계란?



그래서 좋은 SW 설계란?

확장에
열려있으면 됨.



그래서 좋은 SW 설계란?



- S.O.L.I.D 객체지향 5원칙?
- 디자인패턴?



팩트

어차피
설명해도 못
알아먹을거
잖아.



팩트

강 쳐맞아



짜증나는 고객 맛보기



짜증나는 고객 맛보기

1. 초록색 클래스 이름: 본인이 구현했으면 책임 분리 잘 된거임. (확장에 대응 가능)
2. 빨간색 클래스 이름: 새로운 요구사항이 생기면 추가되는 클래스, 없는게 정상.
3. 단순히 초록색 몇개 맞추냐가 설계능력 평가에 절대적 척도가 아님. 그냥 참고사항.



짜증나는 고객 맛보기

각 자동차마다
전진 조건이
다르게
해주세요.
보는 맛이 있게.



각 자동차마다 전진 조건이 달라지게 해주 세요.



○ㅋ, *Car
클래스
생성자에
전진조건
주입해주면 됨.

Code Change Lines: 1

짜증나는 고객 맛보기

ㄴㄴ, 내가
정하는게 아니라
그냥 랜덤으로
알아서
정해주세요.



자동차마다 전진 조건을 랜덤으로 정해주 세요.



○ ㅋ, Car 인스턴스 생성
시점에 `*isMovable()`
메서드가 읽는
`*moveCondition` 필드만
변경하면 됨.

Code Change Lines: 1

- 함수는 하나의 일만 하게 분리하자.
- 단발성으로 사라지는 값이 아니면 필드 변수로 올리자.

짜증나는 고객 맛보기

오, 게임이 더
재밌으면 좋을 거
같은데..
장애물 추가해
주세요.



특정 회차마다 장애물을 등장해 자동차가 전진하지 못하게 해주세요.



○ ㅋ, **Round**
클래스에서
5회차마다
장애물 소환함.

Code Change Lines: 10

특정 회차마다 장애물을 등장해 자동차가 전진하지 못하게 해주세요.



이제부터 자동차들의
`moveRandomly()`
메서드의 성공여부는
Round의 영향을
받아야하네..

특정 회차마다 장애물을 등장해 자동차가 전진하지 못하게 해주세요.



Round가 *Cars의 moveAll()
메서드를 호출하고 있음.
현재 발생한 Event장애물도
moveAll()에 주입해줘야 함.

- 1급 컬렉션:
반복문, 필터링 등을 비즈니스 레이어에서
처리하지마세요.

특정 회차마다 장애물을 등장해 자동차가 전진하지 못하게 해주세요.



만약 Car과 Event가
조우했을 때 action을
Round가 판단해서,
car.move() / car.accident()
메서드를 만들어서
호출하면 안됨.
Car 객체의 책임을
잃어버림

짜증나는 고객 맛보기

오 그럼 장애인물

말고도

아이템도

나오게

해주세요.

마리오카트

마냥



특정 회차마다 아이템을 등장하게 해주세요.



○ ㅋ, **RoundEvent**
라는 인터페이스
추가함.
장애물/랜덤 아이템이
등장할 수 있음.

특정 회차마다 아이템을 등장하게 해주세요.



설계는
이렇게
할거임



인터페이스
RoundEvent
meet(): 이벤트

Implement

구현체
Obstacle
(장애물)
meet(): 이벤트

구현체
RandomItem
meet(): 이벤트

짜증나는 고객 맛보기

(뭐래는거지)



특정 회차마다 아이템을 등장하게 해주세요.



당연히
Round에는
하나의 이벤트만
발생할 수 있음.
ㅇㅋ?

짜증나는 고객 맛보기

아 그런가요..?



짜증나는 고객 맛보기

근데 게임이
너무 재밌네요.
이거 리그전
도입할게요.



짜증나는 고객 맛보기

음, 제가
중단이라고
할때까지 계속
게임
진행해야하고요.



짜증나는 고객 맛보기

게임 기록은
엑셀로
출력해주세요.



1. 중단까지 게임은 계속 반복되어야한다.



음 어지럽네

1. 중단까지 게임은 계속 반복되어야한다.



일단 게임은 **Racing**
클래스로
분리해놔으니까
while문 돌리면 되고..

2. 게임 출력은 엑셀로 뽑아주세요.



어차피
RaceResult
클래스 있으니까
쉽겠네

2. 게임 출력은 엑셀로 뽑아주세요.



*ViewLayer에
ExcelFormatter
를 구현하겠음.

2. 게임 출력은 엑셀로 뽑아주세요.



OutputViewer의
showResult() 메서드
안에 Formmater
호출하면 되겠네.

2. 게임 출력은 엑셀로 뽑아주세요.



도메인에 파싱
로직 넣지마셈.
나중에
제거하거나
변경할 때
힘들어.

짜증나는 고객 맛보기

음 나머지는
내일 할까요



짜증나는 고객 맛보기

전 역시 기획
천재인거
같아요.



사랑스러운 고객과의 조우 끝

좋은 SW 설계

- "시스템을 잘 구조화하면, 새로운 기능을 추가하거나 변경할 때 더 빠르고 효율적으로 작업할 수 있다"
- 어떻게 구조화? -> "Just 짬"
- 그래서 경험을 쌓기 위해 여러분과 함께 라이브 코드 리뷰를 진행하겠습니다.

