



Validate Day02

🕒 Created @October 27, 2021 5:14 PM

Introduction

평가 진행 시 다음과 같은 사항들을 지켜주십시오.

- 상호 간 존중하는 마음으로 예의를 지키며 평가를 진행합니다.
- 평가 진행 중 발생한 기능 장애들은 평가자와 함께 문제점을 밝혀내도록 하십시오. 이 과정에서 토론과 논의가 오갈 수 있습니다.
- 동료의 이해한 것과 다르게 이해할 수도 있다는 점을 배려하십시오. 열린 마음으로 평가에 임하는 것을 권장합니다.

Guidelines

반드시 제출 시 이용되었던 Git 레포지토리를 기준으로 평가를 진행해야 합니다.

평가 진행 시 악의가 있는 별칭 등으로 결과를 조작하지는 않았는지 꼼꼼히 확인하십시오. 그리고 평가 진행 시 이용될 수 있는 스크립트에 대해서도 크로스 체크를 통해 별 문제가 없다는 것을 확인하여 당황스러운 상황을 피할 수 있도록 하십시오.

빈 레포지토리, (서브젝트에서 의도되지 않은) 기능 상에 문제가 있는 프로그램, 설명할 수 없는 코드가 포함된 치팅 등의 상황에서는 **FAIL**을 부여합니다. 특히 치팅에 대해선 경위와 사유를 듣고, 다음에도 같은 상황이 벌어지지 않도록 장려해주시요.

General

- 제출된 레포지토리가 **gofmt**와 **golanci-lint**를 만족하는지 하십시오.

```
gofmt -l [filename]
golanci-lint run [filename]
```

- 제출된 레포지토리에 go.sum이 없거나 변조되어 모듈이 정상 작동하지 않을 수 있습니다. **go mod tidy**를 이용하여 모듈 정리를 진행 했음에도 문제가 있다면, **FAIL**입니다.
- 모든 기능들은 서브젝트에 명시된 *_test.go 파일을 이용하여 제출 이전에 검증되어 있어야 합니다. 테스트 파일이 잘 작성되어 있는지 확인하십시오. 테스트 케이스가 빈약하다면 평가자가 테스트 케이스를 요구할 수 있으며, 제시된 테스트 케이스가 정상적으로 작동하지 않는다면 **FAIL**입니다.

☒ YES

☒ No

Validation

1) ex00

- Product 구조체가 선언되어 있어야 합니다.
- Product 구조체 안에 price, name이 존재해야 합니다.

☒ YES

☒ No

- Product 구조체는 외부에서 접근이 가능해야 합니다.
- Product 구조체의 내부 필드 값은 직접적으로 접근할 수 없어야 합니다.

☒ YES

☒ No

- Product 구조체의 값을 설정하는 메서드가 존재해야 합니다.

- Product 구조체의 값을 설정하는 메서드의 이름이 SetPrice, SetName 과 같은 형태가 아니라면 그 이유에 대해서 확인하십시오. 타당한 이유가 바탕이 되어야 합니다.
- Product 구조체의 값을 가져오는 메서드가 존재해야 합니다.
- Product 구조체의 값을 설정하는 메서드의 이름이 Price, Name 과 같은 형태가 아니라면 그 이유에 대해서 확인하십시오. 타당한 이유가 바탕이 되어야 합니다.

☒ YES

☒ No

- Product를 선언과 동시에 초기화할 수 있도록 해주는 NewProduct 함수를 작성했는지 확인하십시오.
- 다른 패키지에서 NewProduct를 사용하는지, 그 값이 적절히 초기화가 되었는지 확인하십시오.

☒ YES

☒ No

- main.go에서 Product 구조체를 사용해야 합니다.
- main.go의 코드에 대해서 이해가 안되는 부분이 있다면 피평가자는 적절히 설명할 수 있어야 합니다.

☒ YES

☒ No

2) ex01

- Dj 구조체가 선언되어 있어야 합니다.
- Dj 구조체안에 to, from 필드가 선언되어 있어야 합니다.
- Dj 구조체는 Product 구조체를 임베딩하고 있어야 합니다.
- Dj 구조체의 필드는 직접 접근할 수 없어야 합니다.
- 필드를 접근할 수 있도록 메서드를 사용해야 합니다.

☒ YES

☒ No

- Dj 구조체를 선언과 동시에 초기화 할 수 있도록 NewDj 함수가 작성되어 있는지 확인하십시오.
- 다른 패키지에서 NewDj를 사용하는지, 그 값이 적절히 초기화가 되었는지 확인하십시오.

☒ **YES**

☒ **No**

- Info 메서드가 존재해야 합니다.
- Info 메서드를 호출할 때 구조체 안에 빈 필드가 있다면 서브젝트에 명시한 Error 내용으로 출력되는지 확인하십시오.
- Info 메서드에 구조체 안에 빈 필드가 없다면 서브젝트에 명시한대로 출력되어야 합니다.

☒ **YES**

☒ **No**

- Send 메서드가 존재해야 합니다.
- Send 메서드를 호출할 때 구조체 안에 빈 필드가 있다면 서브젝트에 명시한 Error 내용으로 출력되는지 확인하십시오.
- Send 메서드에 구조체 안에 빈 필드가 없다면 서브젝트에 명시한대로 출력되어야 합니다.

☒ **YES**

☒ **No**

3) ex02

- Coubang 구조체가 선언되어 있어야 합니다.
- Coubang 구조체안에 to, from, membership 필드가 선언되어 있어야 합니다.
- Coubang 구조체는 Product 구조체를 임베딩하고 있어야 합니다.
- Coubang 구조체의 필드는 직접 접근할 수 없어야 합니다.
- 필드를 접근할 수 있도록 메서드를 사용해야 합니다.

☒ YES

☒ No

- Dj 구조체를 선언과 동시에 초기화 할 수 있도록 NewDj 함수가 존재해야 합니다.
- 다른 패키지에서 NewDj를 사용하는지, 그 값이 적절히 초기화가 되었는지 확인하십시오.

☒ YES

☒ No

- Info 메서드가 존재해야 합니다.
- Info 메서드를 호출할 때 구조체 안에 빈 필드가 있다면 서브젝트에 명시한 Error 내용으로 출력되는지 확인해보세요
- Info 메서드에 구조체 안에 빈 필드가 없다면 서브젝트에 명시한대로 출력되어야 합니다.

☒ YES

☒ No

- Send 메서드가 존재하는지 확인하십시오.
- Send 메서드를 호출할 때 구조체 안에 빈 필드가 있다면 서브젝트에 명시한 Error 내용으로 출력되는지 확인하십시오.
- Send 메서드에 구조체 안에 빈 필드가 없다면 서브젝트에 명시한대로 출력되어야 합니다.

☒ YES

☒ No

- RocketSend 메서드가 존재하는지 확인하십시오.
- RocketSend 메서드를 호출할 때 구조체 안에 빈 필드가 있다면 서브젝트에 명시한 Error 내용으로 출력되는지 확인하십시오.
- RocketSend 메서드에 구조체 안에 빈 필드가 없다면 서브젝트에 명시한대로 출력되어야 합니다.

☒ YES

☒ No

4) ex03

- Send 메서드가 포함된 Sender 인터페이스를 만들었는지 확인하십시오.

☒ YES

☒ No

- Shipping 함수를 서브젝트에 주어진 형태에 맞게 만들었는지 확인하십시오.
- Shipping 함수를 실행시켰을 때 Type Assertion을 이용하여 Sender 인터페이스가 Coubang 구조체로 인식되는지 확인하십시오.
- Coubang 구조체의 membership 변수의 값이 true라면 RocketSend 메서드를 호출하고, 그렇지 않다면 Send 메서드를 호출하는지 확인하십시오.
- Shipping 함수를 호출하면 Coubang 구조체가 아닌 타입들은 Send 메서드를 호출하는지 확인하십시오.

☒ YES

☒ No

5) ex04

- Type Switch를 이용하여 케이스를 나누어야 합니다. switch가 아닌 if 문으로 작성되어서 매 케이스마다 Type Assertion이 이뤄지면 안 됩니다.
- 서브젝트에 명시한대로 출력이 잘 되는지 확인해야 합니다.
- Send 메서드가 존재하는 새로운 타입을 만들어서 Check 함수에 매개변수로 넣었을 때 "알 수 없는 배송업체입니다." 문구가 출력되는지 확인해야 합니다.

☒ YES

☒ No

6) ex05

- Queue라는 타입이 선언되어 있고 해당 타입으로 변수를 만들 수 있어야 합니다.

☒ YES

☒ No

- ft_Queue 타입으로 변수를 만들고 Push 메서드를 이용하여 데이터를 넣었을 때 값이 들어가는지 확인해야 합니다.
- 서로 다른 여러가지 타입을 넣어보세요. 이전에 만들었던 Coubang, Dj 같은 사용자 지정 타입을 넣어도 좋고 int, string 같은 타입을 넣어도 좋습니다.

☒ YES

☒ No

- Push 후 Pop을 할 때 값을 잘 가져오고 Queue가 첫 번째 인덱스의 값을 삭제하는지 확인해야 합니다.
- ft_Queue가 비어있는데 Pop을 할 경우 어떻게 작동하는지 확인하십시오. 만일 panic 이 발생하여 비정상적인 종료를 할 경우 이후 **No**를 체크하고 이후 항목은 평가하지 않습니다.

☒ YES

☒ No

- ft_Queue를 만들고 Empty 메서드를 실행했을 때, Queue가 비어있는 경우엔 true를 반환하고, Queue 안에 값이 들어있는 경우 false를 반환합니다. 이와 같은 기능이 잘 동작하는지 확인하십시오.

☒ YES

☒ No

- ft_Queue의 데이터를 집어넣고 Size 메서드를 실행시켰을 때, 존재하는 데이터의 수만큼 Size가 반환되는지 확인하십시오.

☒ YES

☒ No

- ft_Queue는 임의의 타입을 모두 저장할 수 있어야 합니다. 예를 들어, 이전에 만든 Coubang, Dj 같은 사용자 지정 타입부터 int, string과 같은 내장 타입까지 저장할 수 있어야 합니다. 그렇지 않으면 No를 체크합니다. 특히 이와 같은 기능이 동작하도록 구현된 부분이 빈 인터페이스를 이용했는지 확인하십시오.

☒ YES

☒ No

Ratings

✓ OK

★ Outstanding project

📁 Empty
work

💾 Invalid
program

👁️ Cheat

💣 Crash