

쉽게 익히는 클린코드

이번주 컨텐츠는?

- ~남새) 긴 매개변수 목록
- *리팩토링) 매개변수를 질의 함수로, 플래그 인수 제거, 여러 함수를 클래스로 묶기

~**نامه**. 긴 매개변수 목록

어떤 함수에 매개변수가 많을수록 함수의 역할을 이해하기 어려워진다.

- a. 과연 그 함수는 한가지 일을 하고 있는게 맞는가?
- b. 불필요한 매개변수는 없는가?
- c. 하나의 레코드로 뭉칠 수 있는 매개변수 목록은 없는가?

[사용 가능한 리팩토링 기술]

객체 통째로 넘기기(03호에서 확인해주세요)

매개변수 객체 만들기(03호에서 확인해주세요)

매개변수를 질의 함수로 바꾸기

플래그 인수 제거하기

여러 함수를 클래스로 묶기

*리팩토링. 매개변수를 질의 함수로 바꾸기

Replace Parameter with Query

함수의 매개 변수 목록은 함수의 다양성을 대변하며, **짧을수록 이해하기 좋다.**

어떤 한 매개변수를 다른 매개변수를 통해 알아낼 수 있다면 '**중복 매개변수**'라 생각할 수 있다.

매개변수에 값을 전달하는 것은 "**함수를 호출하는 쪽**"의 책임이다. 가능하면 함수를 호출하는 쪽의 책임을 줄이고 **함수 내부에서 책임**지도록 노력한다.

"임시 변수를 질의 함수로 바꾸기"
+ "함수 선언 변경하기"
= "매개변수를 질의 함수로 바꾸기"

```
int basePrice = quantity * itemPrice;
double seasonDiscount =
    this.getSeasonalDiscount();
double fees = this.getFees();
double finalPrice =
    discountedPrice(basePrice, seasonDiscount, fees);
```

```
int basePrice = quantity * itemPrice;
double finalPrice
    = discountedPrice(basePrice);
```



*리팩토링. 플래그 인수 제거



플래그는 보통 함수에서 매개변수로 전달해서, 함수 내부의 로직을 분기 하는데 사용한다.

플래그를 사용한 함수는 차이를 파악하기 어렵다.

- *bookConcert(customer, false), bookConcert(customer, true)*
- *bookConcert(customer), premiumBookConcert(customer)*

조건문 분해하기(Decompose Condition)를 활용할 수 있다.

```
void setValues(String name, int value) {
    if(name.equals("height"))
        height = value;
    if(name.equals("width"))
        width = value;
    ...
}
```



```
void setHeight(int arg) {
    height = arg;
}

void setWidth(int arg) {
    width = arg;
}
```



*리팩토링.

여러 함수를 클래스로 묶기

비슷한 매개변수 목록을 여러 함수에서 사용하고 있다면 해당 메소드를 모아서 **클래스**를 만들 수 있다.

클래스 내부로 **메소드를 옮기고**, **데이터를 필드로 만들면** 메소드에 전달해야 하는 매개변수 목록도 줄일 수 있다.

```
double base(Reading reading) { ... }
double taxableCharge(Reading reading) { ... }
double calculateBaseCharge(Reading reading) { ... }
```



```
class Reading {
    double base() { ... }
    double taxableCharge() {...}
    double calculateBaseCharge() {...}
}
```

클린코드 꿀팁을 놓치고 있다면?

**클린코드 팁을
구독하세요:)**

구독하기?!