

1장. 객체, 설계

이번 장 내용 한줄 요약

Tell don't ask와 의존성 관리

src/main/java/racingcar/RacingCarGame.java Outdated

... ... @@ -28,23 +25,47 @@ public void run(int numTrial) {


28 25 }

29 26 }


30 27

28 + public void registerIfWinner(List<String> winners, Car car) {

29 + if (cars.get(0).getPosition() == car.getPosition())

 **javajigi** on 3 Jul 2018 Member + 😊 ...

상태를 가지는 객체의 데이터를 꺼내려(get)하지 말고 객체에 메시지를 보내라는 원칙에 따라 구현한다.

 Reply...

프로그래밍을 제대로 시작하던 순간부터
주구장창 듣던 이야기...

객체지향 생활 체조 총정리

2012/02/03에 게시됨

[소트웍스 앤솔러지](#)에서 발췌한 “객체지향 생활 체조”라는 글을 10차례에 이어 연재했습니다. 지금까지 연재한 9가지 객체지향 훈련 지침을 정리하면 다음과 같습니다.

규칙 1: 한 메서드에 오직 한 단계의 들여쓰기만 한다.

규칙 2: else 예약어를 쓰지 않는다.

규칙 3: 모든 원시값과 문자열을 포장한다.

규칙 4: 한 줄에 점을 하나만 찍는다.

규칙 5: 줄여쓰지 않는다(축약 금지).

규칙 6: 모든 엔티티를 작게 유지한다.

규칙 7: 2개 이상의 인스턴스 변수를 가진 클래스를 쓰지 않는다.

규칙 8: 일급 컬렉션을 쓴다.

규칙 9: 게터/세터/프로퍼티를 쓰지 않는다.

객체지향 생활 체조에도 getter를 사용하지
말고 객체에 메시지를 보내라는 말이 있음

getter를 사용하면 어떻게 되길래?

대충 만든 예제



개발자 1 :

```
public void goForward() {  
    int x = point.getX();  
    x += 1;  
    point.setX(x);  
}  
( 체스말(퀸) 클래스 구현 )
```

대충 만든 예제



개발자 1 :

```
public void goForward() {  
    int x = point.getX();  
    x += 1;  
    point.setX(x);  
}  
( 체스말(퀸) 클래스 구현 )
```

개발자 2 :

```
public void goForward() {  
    int x = point.getX();  
    x += 1;  
    point.setX(x);  
}  
( 체스말(졸따구) 클래스 구현 )
```

대충 만든 예제



개발자 1 :

```
public void goForward() {  
    int x = point.getX();  
    x += 1;  
    point.setX(x);  
}  
( 체스말(퀸) 클래스 구현 )
```

개발자 2 :

```
public void goForward() {  
    int x = point.getX();  
    x += 1;  
    point.setX(x);  
}  
( 체스말(졸따구) 클래스 구현 )
```

·
·
·
·

문제 : 1칸값의 기준이 1에서 2로 바뀜

대충 만든 예제

Point 객체

int x
int y

다 찾아서 고쳐야 땀

체스 게임

goForward()
체스말 전진

개발자 1 :

```
public void goForward() {  
    int x = point.getX();  
    x += 2;  
    point.setX(x);  
}  
( 체스말(퀸) 클래스 구현 )
```

개발자 2 :

```
public void goForward() {  
    int x = point.getX();  
    x += 2;  
    point.setX(x);  
}  
( 체스말(졸따구) 클래스 구현 )
```

·
·
·
·

다 찾아서 고쳐야 됨

->

유지보수 시간이 오래걸림

->

서비스의 성장에 대응하기 어려움

다 찾아서 고쳐야 됨

->

빠뜨린 부분이 있을 수 있음

->

버그 위험성 상승

문제 :

개발자 1,2,3은 $x+2$ 로 구현했지만,
개발자 4는 그것을 인지하지 못하고 $x+1$ 로 개발을 했다.
-> 서비스 정책의 통일성

**getter를 사용해서 다른 객체의 데이터를 가져와 로직을 구현하면
이렇게 절차지향적인 코드가 될 가능성이 높아짐**

근데 절차지향이 꼭 나쁜건가?

 현대자동차 AI Research LAB

 서울 거주

 2006년 10월에 가입

 838명이 팔로우함

 docs.likejazz.com

 함께 하는 항목

 회원님과 상대방 모두 송창현님, 김포프님 외 4명과 친구입니다.

 회원님과 상대방 모두 우아한Tech, 여행을 떠나는 이유 외 페이지 43개를 좋아합니다.

 7월 13일 오후 1:21 · 🌐

영화 예매 시스템이라는 가상의 프로젝트를, 따라하면서 객체지향 설계를 이해할 수 있도록 하는 좋은 책. 흔한 짜깁기가 아닌, 예제 프로젝트를 직접 구현해 나간다는 점에서 높은 점수를 주고 싶으나 procedural programming으로 if else 몇 번이면 구현될 만한 단순한 비즈니스 로직을 왜 이렇게 복잡하게 구현해야 하는지는 여전히 이해하기 어렵다.

오브젝트

OBJECTS

코드로 이해하는
객체지향 설계

IT Leaders .028

조영호 지음



getter를 사용해서 다른 객체의 데이터를 가져와 로직을 구현하면
이렇게 절차지향적인 코드가 될 가능성이 높아짐

(그러나 때론 절차지향적인 코드가 더 나은 경우도 있다. (저자가 말씀하심))

설계는 ‘이 패러다임, 방법론을 사용하면 다 해결돼 !’ 가 아니라
서비스의 상황에 맞추어서
이 상황에선 이렇게 구성해보고, 이걸 써보고, 등 상황에 맞게 바꾸어 나가야한다.

그러므로 프로그래밍은 기술의 구현이 목적이 아니라
서비스의 안정성과 성장이 목적이기 때문에
어느 경우에서나 통하는 만능 원칙이나 기술따위는 없다고 생각한다.
(서비스는 셀 수 없을만큼 종류가 다양하고, 규모 또한 다양하기 때문)

그렇기 때문에 이 분야가 더욱 의식적으로, 혹은 장인정신으로 일에 임해야하는 것 같다.

**이 책은 그러한 변경되는 상황에 대응해야하는 개발을 할 때,
유연하게 코드를 짤 수 있는 방법에 대한 이야기를 하는 것 같다.**

1장과 목차를 읽고난 소감