

아이템16 (public 클래스에서는 public 필드가 아닌 접근자 메서드를 사용하라)

Public 클래스 필드들의 접근자는 public이어서는 안된다.

Public 필드의 단점

1. 데이터 필드에 직접 접근할 수 있어 캡슐화의 이점을 제공하지 못한다.
2. API를 수정하지 않고는 내부 표현을 바꿀 수 없다.
3. 불변식을 보장할 수 없으며 외부에서 필드에 접근할 때 부수 작업을 수행할 수 없다.

Public 필드들은 모두 private으로 바꾸고 public접근자(getter)를 추가하여 사용하자.

```
public class Home {  
  
    public int code;  
    public String address;  
  
}
```

Public 필드를 사용한 클래스

```
public class Home {  
  
    private int code;  
    private String address;  
  
    public int getCode() { return code; }  
  
    public void setCode(int code) { this.code = code; }  
  
    public String getAddress() { return address; }  
  
    public void setAddress(String address) { this.address = address; }  
  
}
```

접근자와 변경자(setter) 메서드를 활용한 캡슐화

패키지 바깥에서 접근할 수 있는 클래스라면 접근자를 제공함으로써 클래스 내부 표현 방식을 언제든지 바꿀 수 있는 유연성을 얻을 수 있다. 만약 public 클래스가 필드를 공개하면 이를 사용하는 클라이언트가 존재할 경우 내부 표현 방식을 마음대로 바꿀 수 없게 된다.

하지만 package-private 클래스 혹은 private 중첩 클래스라면 데이터 필드를 노출해도 문제가 되지 않는다. 어차피 이 클래스를 포함하는 패키지 안에서만 동작하기 때문에 바깥 코드는 손대지 않고 데이터 표현방식을 바꿀 수 있다.

정리

Public 클래스는 절대 가변 필드를 직접 노출해서는 안된다. 불변 필드라면 노출해도 덜 위험하지만 안심할 수는 없다. 하지만 package-private 클래스나 private 중첩 클래스에서는 필드를 노출하는 편이 나을 때도 있다.