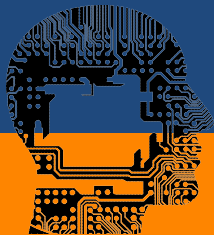




한국IT진흥부설

정보보호교육학원

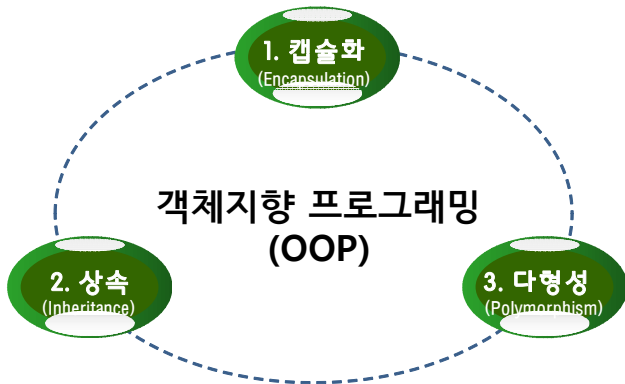
아이섹



자바 프로그래밍 기초

캡슐화

By SoonGu Hong(Kokono)





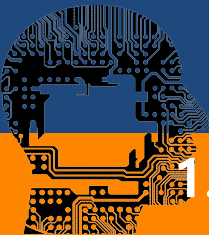
한국IT진흥부설

정보보호교육학원

아이섹

자바 프로그래밍 기초

1. 캡슐화(encapsulation)



1-1. 캡슐화란?

- 캡슐화란 객체의 필드, 메서드를 하나로 묶고, 실제 구현 내용을 감추는 것을 말합니다.
- 외부 객체는 다른 객체의 내부의 구조를 알지 못하며 객체가 의도적으로 노출한 필드와 메서드만 이용할 수 있습니다.
- 캡슐화를 사용하는 이유는 외부의 잘못된 사용으로 인해 객체가 손상되지 않도록 하는데 있습니다.



자동차 부품들을 차체로 캡슐화하여 숨김으로써, 자동차 사용자가 엔진을 건드려서 차량손상을 방지한다!

1-2-1. 캡슐화를 해야 하는 경우

- 일반적으로 객체지향 프로그래밍에서 객체의 속성(필드)은 객체 외부에서 직접적으로 접근하는 것을 막습니다.
- 그 이유는 외부에서 필드에 마음껏 접근할 경우 객체의 무결성이 깨어질 수 있기 때문입니다.
- 예를들어 고양이의 몸무게는 음수가 될 수 없는데 외부에서 필드 값을 음수로 변경해버리면 객체의 무결성이 깨지기 때문입니다.

```
public class Cat {  
    double weight;  
}
```



```
Cat cat = new Cat();  
cat.weight = -99;
```



이런 코드는 막아야 한다!

1-2-2. 캡슐화 방법

- 객체의 무결성을 지키기 위해 필드는 철저히 감추는 것이 좋습니다. 그렇게 하기 위해 접근 제한자를 이용합니다.
- `private` 제한을 필드에 걸어서 외부 객체가 필드에 직접 접근할 수 없게 합니다.

```
public class Cat {  
    private double weight;  
}
```



```
Cat cat = new Cat();  
cat.weight = -99; (X)
```

↑
weight에 private이 걸려
이 코드는 컴파일 에러가
발생한다.

1-3-1. getter와 setter

- 접근제한자 `private`을 사용하여 필드접근을 차단한 것까지는 좋은데 문제는 외부에서 해당 필드값을 제어할 수단이 사라졌다는 것입니다.
- 캡슐화의 의도는 필드 완전 접근 차단보다는 객체의 무결성을 해치는 값 저장을 방지하는 것이 목적이므로 메서드를 통해 제약조건을 설정하여 필드를 제어할 수 있도록 합니다.
- 메서드를 사용하면 매개값을 통한 값 검증로직을 구성할 수 있기 때문에 객체지향프로그래밍에서는 메서드를 통한 필드값 제어를 선호합니다.
- 이 때 값 변경을 위해 사용하는 메서드를 **setter**라고 부르며 값 참조를 위해 사용하는 메서드를 **getter**라 부릅니다.

1-3-2. setter

- setter는 객체의 필드값을 변경할 때 사용하는 메서드입니다.
- 외부에서 접근하여 필드값을 변경할 수 있도록 접근 제한자는 **public**으로 설정합니다.
- 매개변수로 필드에 초기화할 값을 받은 뒤 해당 값을 검증한 뒤 필드에 초기화합니다.
- 관례상 setter의 이름은 **set + 필드명(첫글자대문자)**로 지정합니다.

```
public class Cat {  
    private double weight;  
    public void setWeight(double weight) { //setter  
        if(weight <= 0) {  
            this.weight = 0;  
            return;  
        }  
        this.weight = weight;  
    }  
}
```


1-3-3. getter

- getter는 객체의 필드값을 참조할 때 사용하는 메서드입니다.
- 외부에서 접근하여 필드값을 참조할 수 있도록 접근 제한자는 public으로 설정합니다.
- 외부에 필드값을 전달할 때 값을 형식에 맞게 바꾸어줄 수도 있습니다.
- 관례상 getter의 이름은 get + 필드명(첫글자대문자)로 지정합니다.
- 필드의 타입이 boolean인 경우 이름은 is + 필드명으로 지정합니다.

```
public class Cat {  
    private double weight; //저장값 단위 파운드  
    public void setWeight(double weight) { //setter  
        ...  
        public double getWeight() { //getter  
            return this.weight * 0.45; //kg변환해서 리턴  
        }  
    }  
}
```

감사합니다
THANK YOU