

우리가 쓰고 있는 **Person** 클래스의 매우 간소화된 버전입니다:

Person.java

```
public class Person {  
    int age;  
}
```

Driver.java

```
public class Driver {  
    public static void main(String[] args) {  
        Person p1 = new Person();  
        p1.age = 28;  
        System.out.println(p1.age);  
    }  
}
```

그런데 사실 이걸 굉장히 위험한 코드입니다. **p1.age = 28;** 을 **p1.age = -10;** 으로 바꾸어도 아무 문제 없이 실행되죠? 나이가 음수인 건 말이 안 되는데 말이죠...

접근 제어자

방금 보신 것처럼, 객체에 대한 권한을 너무 쉽게 주면 문제가 생길 수 있습니다. 다행히 자바에서는 권한을 제한해주는 '접근 제어자(Access Modifier)'라는 게 있습니다.

Person.java

```
public class Person {  
    private int age;  
}
```

인스턴스 변수 앞에 **private** 이라는 접근 제어자를 써주면 그 변수는 해당 클래스 내에서만 접근 가능합니다. 이제 **Driver** 는 **age** 에 대한 권한이 없기 때문에 **Driver** 에서 **p1.age = -10;** 을 해주려고 하면 오류가 나올 것입니다. 마찬가지로 **System.out.println(p1.age);** 도 불가능합니다.

그러면 이제는 **age** 에 대한 권한이 아예 없는 걸까요?

아닙니다! **private** 변수라도 그 클래스 내에서는 사용이 가능하기 때문에, 그 클래스에 **age** 를 사용하는 메소드를 쓰면 됩니다.

Person.java

```

public class Person {
    private int age;

    public void setAge(int newAge) {
        if (newAge > 0) {
            age = newAge;
        }
    }

    public int getAge() {
        return age;
    }
}

```

이제 **setAge** 와 **getAge** 메소드를 이용해서 **age** 변수를 바꿔주고 받아올 수 있습니다. **setAge** 와 **getAge** 같은 메소드를 흔히 'setter' 메소드와 'getter' 메소드라고 부릅니다.

그런데 보시면 두 메소드 앞에는 **public** 이라는 접근 제어자를 써주었죠? 변수 **age** 는 숨기고 싶어서 **private** 으로 설정했지만, **setAge** 와 **getAge** 는 다른 클래스에서도 사용하고 싶기 때문에 어디서든 접근 가능하게 해주는 **public** 을 썼습니다.

이제 드라이버를 이렇게 바꿀 수 있습니다:

Driver.java

```

public class Driver {
    public static void main(String[] args) {
        Person p1 = new Person();
        p1.setAge(28);
        p1.setAge(-10);
        System.out.println(p1.getAge());
    }
}

```

28



수업을 완료하셨으면 체크해주세요.



수강생 Q&A 보기



(/questions? 질문하기

assignment_id=346&sort_by=popular)
(/questions/new?

assignment_id=346&op1=%EA%B0%9D%EC%B2%B4+%EC%A7

< 이전 강의
접근 제어자 (/assignments/327)

다음 강의
접근 제어자 연습 > (/assignments/334)