

```
1 package dataHiding17_02;
2
3 public class Student {
4
5     private String name;
6     private int score;
7
8     public Student(String n, int s) {
9         this.name = n;
10        this.score = s;
11    }
12
13    public void getInfo() {
14        System.out.println("-- getInfo() --");
15        System.out.println("name: " + this.name);
16        System.out.println("score: " + this.score);
17    }
18
19    // getters, setters 를 만든다. Code > Generate > (choose) Getter and Setter.
20    // private으로 막아둔 속성에 접근할 수 있다.
21
22    public String getName() {
23        return name;
24    }
25
26    // 이름은 바꾸면 안 되므로, setName() 은 지워도 된다.
27    // public void setName(String name) {
28    //     this.name = name;
29    // }
30    // 이제 한 번 초기화된 name 을 바꿀 수 있는 방법은 영구히 사라졌다.
31    // 웹 사이트 가입 후 아이디는 변경이 안 된다. 이럴 때 사용한다.
32
33    public int getScore() {
34        return score;
35    }
36
37    // 왜 이런 getter와 setter 같은 방법을 사용하는가?
38    // 안전 장치를 만들 수 있다.
39    // score가 50 미만으로 들어오면 변경할 수 없다.
40    public void setScore(int score) {
41        if (score > 50) this.score = score;
42    }
43
44    // 급여가 얼마 이하면, 최저 임금 이하로는 줄 수가 없다는 조건을 달 때도 사용할 수 있다.
45    // 최저 임금 이하로 들어오면 경고를 띄워서 다시 조정할 수 있게 한다.
46
47 }
48
```

```
1 package dataHiding17_02;
2
3 public class MainClass {
4     public static void main(String[] args) {
5         Student student1 = new Student("Hong Gildong", 90);
6         student1.getInfo();
7
8         // 홍길동이 공부를 열심히 해서 100점이 됐다.
9         // private 이므로 속성을 바꿀 수 없다.
10        // setter 메서드를 사용하면, 바꿀 수 있다.
11
12        student1.setScore(100);    // 홍길동의 점수를 바꿀 수 있다.
13        student1.getInfo();
14
15        student1.setScore(40);    // 점수는 변경되지 않는다.
16        student1.getInfo();
17
18    }
19 }
20
```