

16.1 Encapsulation

Encapsulation is to make sure that "sensitive" data is hidden from users. To achieve this:

- declare class variables/attributes as **private**
- provide public **get** and **set** methods to access and update the value of a **private** variable
- **private** variables can only be accessed within the same class (an outside class has no access to it). However, it is possible to access them if we provide public **get** and **set** methods.

Example

```
1 public class Animal {
2     private String name; // private = restricted access
3
4     // Getter
5     public String getName() {
6         return name;
7     }
8
9     // Setter
10    public void setName(String newName) {
11        this.name = newName;
12    }
13 }
```

💡 Why Encapsulation?

- Better control of class attributes and methods
- Class attributes can be made **read-only** (if you only use the **get** method), or **write-only** (if you only use the **set** method)
- Flexible: the programmer can change one part of the code without affecting other parts
- Increased security of data.

How to use encapsulated class?

```
1 Animal a = new Animal();
2 a.setName("Tiger");
3
4 System.out.println(a.getName());
```

code by **Suyash** 😊