🔐 Encapsulation in Java

✅ What is Encapsulation?

Encapsulation is a fundamental concept in object-oriented programming (OOP) that **bundles data (variables) and methods (functions)** that operate on the data into a single unit called a **class**. It also **restricts direct access** to some of the object's components, which is a means of **data hiding**.

🎯 Benefits of Encapsulation

- Protects the internal state of an object.
- Improves code maintainability.
- Allows control over how fields are accessed and modified.
- Makes the code more flexible and extensible.

🧰 How to Achieve Encapsulation in Java

1. Declare the variables `private`.
2. Provide `public` getter and setter methods to access and update the values.

💡 Example: Student Class

```
// Class definition
public class Student {
    // Step 1: Private fields
    private String name;
    private int age;

    // Step 2: Public Getter for name
    public String getName() {
        return name;
    }
```

```
    // Step 3: Public Setter for name
    public void setName(String newName) {
        name = newName;
    }

    // Getter for age
    public int getAge() {
        return age;
    }

    // Setter for age with a validation check
    public void setAge(int newAge) {
        if (newAge > 0) {
            age = newAge;
        } else {
            System.out.println("Age must be positive.");
        }
    }
}
```

🧪 Test the Class in `Main` Method

```
public class Main {
    public static void main(String[] args) {
        Student student = new Student();

        student.setName("Rahul");
        student.setAge(20);

        System.out.println("Name: " + student.getName());
        System.out.println("Age: " + student.getAge());
    }
}
```

📝 Output:

```
Name: Rahul
Age: 20
```

✅ Summary

Encapsulation keeps your class's internal data **safe and hidden** from external misuse. You control access to fields using getter and setter methods.
