**1. Tính đóng gói là gì? Tại sao tính đóng gói lại quan trọng trong lập trình đối tượng?**

Tính đóng gói (Encapsulation) là một nguyên tắc quan trọng trong lập trình hướng đối tượng, giúp che giấu dữ liệu bên trong một lớp và chỉ cho phép truy cập dữ liệu thông qua các phương thức được chỉ định (getter, setter).

**Tầm quan trọng của tính đóng gói:**

* **Bảo vệ dữ liệu:** Ngăn chặn việc truy cập hoặc thay đổi dữ liệu trực tiếp từ bên ngoài, giúp tránh lỗi và bảo mật tốt hơn.
* **Kiểm soát quyền truy cập:** Chỉ định rõ ràng dữ liệu nào có thể được truy cập từ bên ngoài thông qua các phương thức phù hợp.
* **Dễ bảo trì và mở rộng:** Khi cần thay đổi cách hoạt động của lớp, ta chỉ cần chỉnh sửa bên trong mà không ảnh hưởng đến các phần khác của chương trình.
* **Tăng tính linh hoạt:** Có thể kiểm tra và hợp thức hóa dữ liệu trước khi thay đổi nó.

**2. Làm thế nào để áp dụng tính đóng gói trong Java?**

Trong Java, ta có thể áp dụng tính đóng gói bằng cách:

1. **Khai báo các thuộc tính (biến) của lớp là private**, để ngăn không cho truy cập trực tiếp từ bên ngoài.
2. **Cung cấp các phương thức public để truy xuất và thay đổi giá trị dữ liệu** thông qua getter (để lấy dữ liệu) và setter (để thay đổi dữ liệu).
3. **Có thể thêm logic kiểm tra dữ liệu** trong setter để đảm bảo dữ liệu hợp lệ.

**Getter và Setter trong Java:**

* **Getter** là phương thức trả về giá trị của một thuộc tính.
* **Setter** là phương thức gán giá trị mới cho một thuộc tính, có thể kèm theo kiểm tra hợp lệ.

**3. Ví dụ về tính đóng gói trong Java**

public class Person {

private String name;

private int age;

public Person(String name, int age) {

this.name = name;

this.age = age;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

if (age > 0) {

this.age = age;

} else {

System.out.println("Tuổi phải lớn hơn 0!");

}

}

public void display() {

System.out.println("Tên: " + name + ", Tuổi: " + age);

}

}

public class Main {

public static void main(String[] args) {

Person person = new Person("Nguyễn Văn A", 25);

System.out.println("Tên: " + person.getName());

System.out.println("Tuổi: " + person.getAge());

person.setName("Trần Thị B");

person.setAge(-5); // Dữ liệu không hợp lệ, sẽ bị từ chối

person.display();

}

}

**4. Lợi ích của tính đóng gói trong bảo vệ dữ liệu và kiểm soát quyền truy cập**

* **Bảo mật dữ liệu:** Ngăn chặn việc thay đổi dữ liệu một cách tùy tiện từ bên ngoài.
* **Kiểm soát dữ liệu hợp lệ:** Các setter có thể thêm điều kiện kiểm tra để đảm bảo dữ liệu hợp lệ.
* **Giảm rủi ro lỗi:** Hạn chế thay đổi dữ liệu sai sót từ bên ngoài, giúp chương trình hoạt động ổn định.
* **Tăng khả năng bảo trì và mở rộng:** Khi cần thay đổi cách hoạt động của lớp, chỉ cần cập nhật trong nội bộ lớp mà không ảnh hưởng đến mã bên ngoài.