

1. Lợi ích của Tính Đa Hình (Polymorphism) trong Java

Polymorphism cho phép bạn xử lý các đối tượng thuộc các lớp khác nhau thông qua cùng một **interface hoặc lớp cha**. Những lợi ích chính:

- **Tăng tính linh hoạt:** Bạn có thể viết code chung cho nhiều loại đối tượng khác nhau.
- **Giảm sự phụ thuộc giữa các lớp:** Giao tiếp qua interface hoặc lớp trừu tượng, dễ thay đổi hoặc mở rộng chương trình.
- **Tăng khả năng mở rộng và bảo trì:** Khi thêm lớp mới, bạn không cần sửa đổi code đã có.
- **Dễ dàng sử dụng lại code:** Các hành vi giống nhau có thể dùng chung logic, nhờ override phương thức.

2. Kế thừa giúp đạt được Đa hình như thế nào?

Inheritance (kế thừa) là nền tảng để thực hiện đa hình. Khi một lớp con kế thừa từ lớp cha:

- Nó kế thừa các **phương thức và thuộc tính**.
- Nó có thể **ghi đè (override)** các phương thức của lớp cha.
- Khi dùng kiểu dữ liệu của lớp cha, bạn vẫn có thể xử lý đối tượng lớp con (runtime polymorphism).

3. So sánh Polymorphism và Inheritance

TIÊU CHÍ	POLYMORPHISM	INHERITANCE
KHÁI NIỆM	Một hành vi có thể có nhiều hình thái	Lớp con kế thừa thuộc tính và phương thức từ lớp cha
MỤC ĐÍCH	Tăng tính linh hoạt, cho phép dùng chung interface hoặc lớp cha	Tái sử dụng code, tổ chức phân cấp lớp
CÁCH HOẠT ĐỘNG	Qua override hoặc overload phương thức	Qua extends hoặc implements
LOẠI	Compile-time (overloading) và Runtime (overriding)	Single, multilevel, hierarchical inheritance
QUAN HỆ	Thường dựa trên inheritance để thực hiện	Là cấu trúc phân cấp của các lớp
VÍ DỤ THỰC TẾ	Gọi draw() trên nhiều loại Shape	Circle kế thừa từ Shape