

1. Những lợi ích của Tính đa hình (Polymorphism):

- **Tăng tính linh hoạt và mở rộng:** Cho phép sử dụng cùng một phương thức với các đối tượng khác nhau, giúp chương trình dễ mở rộng và bảo trì.
- **Giảm sự phụ thuộc giữa các thành phần:** Giúp lập trình viên viết mã ít phụ thuộc vào lớp cụ thể, dễ dàng thay đổi hoặc mở rộng chức năng mà không ảnh hưởng đến các phần khác.
- **Tăng khả năng tái sử dụng mã:** Việc sử dụng một phương thức chung cho nhiều đối tượng giúp tiết kiệm công sức viết lại mã.
- **Dễ bảo trì và kiểm thử:** Do mã được tổ chức rõ ràng, sử dụng interface hoặc lớp cha làm chuẩn, nên dễ dàng sửa lỗi và kiểm tra.

2. Kế thừa (Inheritance) có vai trò như thế nào trong việc đạt được Tính đa hình trong Java?

- Trong Java, kế thừa cho phép một lớp con kế thừa các thuộc tính và phương thức của lớp cha.
- Khi lớp con ghi đè (override) các phương thức từ lớp cha, ta có thể sử dụng đối tượng lớp con thông qua tham chiếu của lớp cha, điều này chính là cơ sở để đạt được tính đa hình.
- Ví dụ: nếu Animal là lớp cha và Dog, Cat là lớp con, ta có thể viết Animal a = new Dog(); hoặc Animal a = new Cat(); và gọi a.sound() — mỗi lớp con sẽ có cách thực hiện sound() riêng.

3. Sự khác nhau giữa Tính đa hình (Polymorphism) và Kế thừa (Inheritance) trong Java:

Đặc điểm	Kế thừa (Inheritance)	Tính đa hình (Polymorphism)
Mục đích	Tái sử dụng mã và mở rộng chức năng	Cho phép một hành vi có nhiều cách thực hiện khác nhau
Quan hệ	Thiết lập mối quan hệ cha - con giữa các lớp	Cho phép đối tượng có nhiều hình thức khác nhau
Tính năng chính	Kế thừa thuộc tính và phương thức từ lớp cha	Ghi đè (override) và đa hình thời gian chạy
Phạm vi áp dụng	Chủ yếu trong cấu trúc lớp	Trong cả lớp và giao diện (interface)