

Polymorphism and Inheritance in Java

1. What are the advantages of Polymorphism?

Tính linh hoạt (Flexibility): Polymorphism cho phép xử lý nhiều loại đối tượng khác nhau thông qua cùng một interface hoặc lớp cha. Nhờ đó, ta có thể viết code tổng quát, ví dụ thao tác với `ArrayList<Media>` mà không cần quan tâm đến đối tượng cụ thể là `Book`, `CD` hay `DVD`.

Tính mở rộng (Extensibility): Giúp thêm các lớp con mới mà không cần sửa đổi logic xử lý đã có. Khi lớp mới kế thừa lớp cha, chương trình vẫn có thể sử dụng đối tượng mới này như một phần của hệ thống cũ.

Tái sử dụng mã (Reusability): Cho phép sử dụng lại mã từ lớp cha, nhờ đó tránh việc viết lại logic xử lý cho từng lớp con cụ thể.

Code dễ đọc và dễ quản lý hơn: Giảm bớt sự phụ thuộc vào các cấu trúc điều kiện như `if-else` hoặc `switch-case` khi xử lý đối tượng theo kiểu cụ thể.

2. How is Inheritance useful to achieve Polymorphism in Java?

Kế thừa là cơ chế then chốt để thực thi đa hình trong Java. Thông qua mối quan hệ "is-a" giữa lớp con và lớp cha, ta có thể khai báo biến tham chiếu kiểu lớp cha và gán đối tượng thuộc lớp con cho nó. Khi gọi phương thức bị override, Java sẽ xác định và gọi đúng phiên bản phương thức tương ứng của lớp con tại thời điểm chạy (late binding).

3. What are the differences between Polymorphism and Inheritance in Java?

Inheritance (Kế thừa): Là cơ chế cho phép lớp con kế thừa các thành phần (fields, methods) từ lớp cha. Inheritance cho phép tái sử dụng mã và thiết lập mối quan hệ phân cấp giữa các lớp. Đây là nền tảng để thực hiện được đa hình.

Polymorphism (Đa hình): Là khả năng thực thi phương thức theo những cách khác nhau tùy thuộc vào đối tượng cụ thể. Đa hình trong Java thường được thực hiện thông qua ghi đè phương thức (method overriding) và liên kết động (dynamic binding).