

1. Lợi ích của Polymorphism trong Java là gì?(What are the advantages of Polymorphism?)

Polymorphism (đa hình) là tính chất cho phép các đối tượng thuộc những lớp khác nhau có thể được xử lý thông qua cùng một giao diện (interface) hoặc lớp cha. Trong Java, điều này chủ yếu xảy ra khi một lớp con override lại phương thức của lớp cha.

Lợi ích cụ thể:

a. Giảm phụ thuộc vào kiểu cụ thể

- Bạn có thể sử dụng lớp cha làm kiểu dữ liệu cho các biến, mảng, danh sách, đối số phương thức, trả về kết quả,...
- Ví dụ:

List<Media> mediaList = new ArrayList<>();
mediaList.add(new Book(...));
mediaList.add(new CompactDisc(...));

b. Cho phép mở rộng dễ dàng

- Bạn có thể thêm các lớp con mới mà không cần chỉnh sửa code hiện tại. Ví dụ: thêm một lớp mới VinylDisc vẫn hoạt động được trong List<Media> mà không sửa logic xử lý.

c. Tăng tính linh hoạt của chương trình

- Gọi cùng một phương thức nhưng hành vi thực thi khác nhau, tùy thuộc vào đối tượng thực.
- Ví dụ: media.play() sẽ gọi đúng play() của CompactDisc hoặc DigitalVideoDisc nếu có override.

d. Hỗ trợ clean code và giảm các câu lệnh điều kiện

- Không cần if-else hay switch-case để phân biệt kiểu đối tượng.
- Tất cả hành vi xử lý được "đẩy" về các lớp con để thực hiện đúng chức năng.

2. Kế thừa giúp đạt được đa hình trong Java như thế nào?(How is Inheritance useful to achieve Polymorphism in Java?)

Inheritance (Kế thừa) là cơ sở quan trọng để đạt được Polymorphism.

a. Kế thừa cho phép định nghĩa hành vi chung trong lớp cha

- Lớp Media có thể định nghĩa phương thức như toString(), play(), getTitle()...
- Các lớp con như Book, DigitalVideoDisc, CompactDisc **kế thừa các phương thức này**.

b. Lớp con có thể override phương thức của lớp cha

- Mỗi lớp con cung cấp phiên bản cụ thể của toString() hay play().
- Điều này giúp đảm bảo khi bạn gọi media.play(), **Java sẽ xác định đúng phương thức override tại runtime** (đa hình động).

Ví dụ minh họa:

Media media = new DigitalVideoDisc(...);
media.play(); // Gọi play() của DigitalVideoDisc, không phải của Media

→ Đây là **tính đa hình thông qua kế thừa và override**.

3. Sự khác biệt giữa Polymorphism và Inheritance trong Java là gì?(What are the differences between Polymorphism and Inheritance in Java?)

So sánh	Inheritance (Kế thừa)	Polymorphism (Đa hình)
Định nghĩa	Cơ chế cho phép lớp con kế thừa thuộc tính và phương thức của lớp cha.	Cơ chế cho phép gọi cùng một phương thức nhưng hành vi phụ thuộc vào kiểu đối tượng thực tế.
Vai trò	Tái sử dụng mã nguồn (reuse code) và mô hình hóa mối quan hệ “is-a”.	Tùy biến hành vi theo lớp con mà không thay đổi code gọi.
Thực hiện	Dùng extends hoặc implements.	Dùng override, dynamic binding và kiểu tham chiếu lớp cha.
Ví dụ	class DVD extends Media	media.toString() → in thông tin đúng theo loại DVD, CD hoặc Book
Tính linh hoạt	Cứng hơn, yêu cầu quan hệ rõ ràng giữa lớp cha và lớp con.	Mềm dẻo hơn, tận dụng override để mở rộng mà không sửa đổi code cũ.