**6. Inheritance**

|  |  |
| --- | --- |
| Tính kế thừa?  Đặc điểm | Tính kế thừa là cơ chế cho phép 1 lớp con sử dụng lại các đặc điểm hành vi đã được định nghĩa trước đó ở lớp cha.  Mục đích: tái sử dụng mã nguồn.  Đặc điểm:   * Lớp con kế thừa nhưng vẫn đảm bảo quy tắc access modifier * Không có kế thừa constructor * Java không hỗ trợ đa kế thừa * Mối quan hệ kế thừa là is-a |
| Từ khóa super và final | super: sử dụng ở lớp con dùng để gọi constructor, method, thuộc tính ở lớp cha.  Dùng final cho class để tạo lớp không có lớp con. |
| Lớp Object | Là lớp gốc của tất cả các lớp trong Java.  toString() là phương thức trả về 1 chuỗi biểu diễn thông tin của đối tượng. |
| Tính đa hình | Là 1 đối tượng có thể hiện/ hành vi theo nhiều cách khác nhau phụ thuộc vào ngữ cảnh.   |  |  | | --- | --- | | Overriding(Ghi đè) | Overloading(Nạp chồng) | | Overriding là quá trình định nghĩa một phương thức trong lớp con (subclass) có cùng tên, cùng kiểu trả về và cùng danh sách tham số với phương thức trong lớp cha (superclass). | Overloading là quá trình định nghĩa nhiều phương thức cùng tên trong một lớp, nhưng với số lượng hoặc kiểu tham số khác nhau. | | Phương thức Overriding phải cùng tên, cùng danh sách tham số, cùng kiểu trả về |  | | Access modifier ở phương thức con phải có level bằng hoặc cao hơn so với phương thức lớp cha |  | | Dùng final để cấm ghi đè phương thức ở lớp cha |  | | Xảy ra ở class có mối quan hệ kế thừa is-a | Xảy ra trong 1 class | | Đa hình tại run-time | Đa hình tại compile | |
| Ép kiểu | Ép kiểu là cơ chế chuyển đổi 1 tham chiếu đến đối tượng thuộc loại này thành tham chiếu đến đối tượng thuộc loại khác  Có 2 loại ép kiểu:  Ép kiểu ngầm định(Implicitcasting): ép từ subtype lên supertype  Ép kiểu tường minh(Explicitcasting): ép từ supertype xuống subtype |