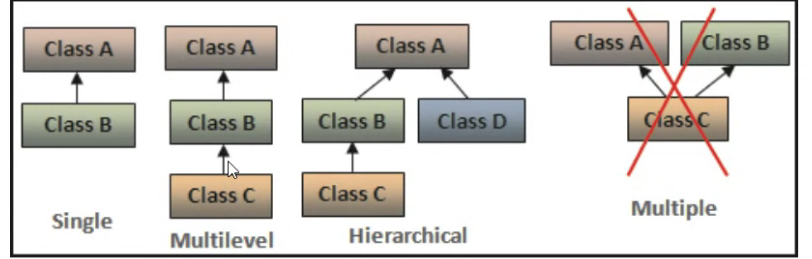
SESSION 6 – Kế thừa và tính đa hình của OOP

1. Khái niệm về kế thừa

* Kế thừa là cơ chế cho phép class con sử dụng lại các thuộc tính và hành vi đã được định nghĩa ở class cha mà không cần khai báo lại.
* Mục đích của kế thừa : tái sử dụng code
* Mối quan hệ kế thừa giữa lớp cha và lớp con : is -a
  + Lớp cha thường được gọi là base class, super class
  + Lớp con thường được gọi là : child class , subclass
* Lớp con không thể truy cập các thuộc tính private của lớp cha
* Lớp con không thể kế thừa constructor của lớp cha
* Java không hỗ trợ đa kế thừa ( một con không có nhiều lớp cha



* Có thể gọi constructor của class cha thông qua từ khóa super
* Từ khóa final dùng để quy định 1 class không cho kế thừa
* Từ khóa final dùng để quy định 1 phương thức không được cho phép overriding

1. Overriding và overloading

|  |  |
| --- | --- |
| Overriding (ghi đè) | Overloading |
| Là cơ chế cho phép lớp con định nghĩa lại nội dung các method đã có trước đó ở lớp cha  \_Method overriding phải cùng tên, cùng danh sách tham số, cùng kiểu trả về  \_Access modifier phải bằng hoặc cao hơn access modifier của class cha | Là cơ chế cho phép 1 class có khả năng định nghĩa ra nhiều phương thức cùng tên nhưng khác nhau về tham số |
| Xảy ra tại 2 class có mối quan hệ kế thừa (is – a) | Xảy ra nội bộ bên trong một class |
| Thời điểm xảy ra đa hình là runtime | Thời điểm xảy ra đa hình là compile |
|  |  |

1. Dymamic Binding

* Là cơ chế cho phép khai báo dư liệu là class nhưng tham chiếu đến đối tượng là con

1. Ép kiểu trong java

* Có 2 loại ép kiểu :
  + Implicicasting(Ép kiểu ngầm định) : ép từ subtype lên supertype
  + Explicitcasting(ép kiểu tường minh) : ép từ supertype xuống subtype