Abstract class và Interface 1

# Abstract class và Interface

          Apple vừa đọc 1 thread, cũng nêu ra 1 câu hỏi apple đã từng gặp trong khi phỏng vấn xin việc “ phân biệt 1 class abstract & 1 class interface, chúng khác nhau ở điểm nào”.

       Lúc đó cố gắng lắm cũng chỉ trả lời được class abstract là để mô tả lớp trừu tượng hay lớp cha và từ đó để cho các lớp con kế thừa các thuộc tính và hành vi bao gồm các biến và phương thức. Các lớp con extends có thể mở rộng các phương thức của lớp cha. Còn class interface chỉ là giao diện, cung cấp các phương thức trừu tượng, chỉ nêu ra tên phương thức, nhưng không có phần thân và để cho lớp con kế thừa lại các biến và phương thức. Các lớp con implements các lớp interface phải kế thừa lại toàn bộ các phương thức đã được định nghĩa ở lớp interface trên. Đến khi người ta hỏi lại “vậy đâu là điểm khác nhau cơ bản của chúng, nếu chỉ như vậy thì có cần phân biệt chúng ra không hay việc sử dụng là như nhau”. Apple im re luôn không nghĩ ra câu trả lời, đến khi zề rồi mới lục lọi tìm tòi. Hôm nay gặp thread gần gần như vậy nên tổng hợp lại thông tin share với mọi người:

**So sánh class abstract và class interface trong java**

**Giống:**  
1. Không cho phép tạo instance

2. Đều hỗ trợ tính đa hình  
  
**Khác:**  
**1. Hiện thực thân method:**   
[interface]: Không cho phép thực hiện phần thân method    
[abstract]: Có thể thực hiện phần thân method trong đó  
  
**2. Khai báo biến thành viên:**   
[interface] : Không được  
[abstract] : Được  
  
**3. Tầm vực**  
[interface] : Tất cả method là public  
[abstract]: Không ràng buộc, các method, biến thành viên có thể là public/ private/ protected, hoặc abstract  
  
**4. Từ khóa kế thừa của lớp con**  
[interface]: Sử dụng từ khóa "implements" để thừa kế  
[abstract]: Sử dụng từ khóa "extends" để thừa kế

Để hiểu sâu hơn có thể  dịch 2 từ khóa này như sau:

implements nghĩa là ta triển khai 1 phương thức đã được định nghĩa.

extends nghĩa là mở rộng 1 phương thức. (phương thức này đã được định nghĩa và đã có xử lý của nó, ta chỉ mở rộng thêm cho nó mà thôi).

**5.Thừa kế**  
[interface]: Không thể thừa kế từ 1 **interface** khác  
[abstract]: Có thể thừa kế từ 1 lớp **abstract** khác

Note: =>  nói rõ hơn là:  
         +             1 class extends 1 abstract class  
         + 1 class implements 1 interface  
         + 1 interface extends 1 interface  
         + 1 interface không thể implements 1 interface

[interface]: Không thể thừa kế từ 1 lớp **abtract**  
[abstract]: Có thể thừa kế từ 1 **interface**

[interface]: Cho phép đa thừa kế

[abstract]: không cho phép đa thừa kế.

# Sự khác nhau giữa Tablemap và Hashmap

# Collections và List