**Class:**

Lớp (Class) trong lập trình hướng đối tượng là một khuôn mẫu hoặc một bản thiết kế, mô hình mô phỏng các đối tượng.

Đối tượng là các thể hiện của lớp, một đối tượng là một sự tượng trưng cho một thực thể, hoặc là thực thể tồn tại trong thế giới đời thực hoặc thực thể mang tính khái niệm. Một đối tượng có thể tượng trưng cho cái gì đó cụ thể.

Ví dụ: một chiếc Mercedes Benz là một đối tượng thuộc lớp Car, một chiếc IphoneX là đối tượng thuộc lớp Mobile,…

Các ngôn ngữ hướng đối tượng như Java, C#,… hỗ trợ khái niệm kế thừa. Kế thừa có nghĩa là một đối tượng có thể được thừa kế các hành vi và phương thức của lớp. Bất cứ khi nào bạn tạo đối tượng mới của một lớp, bạn nói rằng bạn đang khởi tạo nó.

**Lớp trừu tượng (Abstract Class):**

Là một lớp nhưng không thể tạo ra thực thể. Abstract class có thể chứa hoặc không chứa phương thức abstract – một phương thức chỉ có khai báo mà không chứa cài đặt.

Abstract: trừu tượng >< Final (cuối cùng, không thể kế thừa) nên ta hiểu abstract là chưa xong, chưa hoàn thành, phải có kế thừa…

## Interface:

Một interface không phải là một lớp (class), giống như lớp trừu tượng. Nó chứa các phương thức mà không phần thân. Một interface không thể tự làm bất cứ điều gì. Interface như một khuôn mẫu. Với ngôn ngữ Java không hỗ trợ đa kế thừa nhưng có thể triển khai nhiều interface.

## Sự khác biệt giữa Abtract class và Interface

Một interface chứa một tập hợp các phương thức. Một lớp implements interface phải triển khai các phương thức này.

Một lớp trừu tượng, giống như một interface, sẽ chứa các phương thức. Tuy nhiên, sẽ có ít nhất một phương thức đã hoàn thành tức là phương thức có phần thân. Đây là một sự khác biệt chính giữa abstract class và interface

**Ngoài sự khác biệt lớn này, đây là một số khác biệt giữa Abstract class và Interface**

***Theo cách ngắn gọn, quan hệ giữa một class khi thừa kế một abstract class được gọi là is-a, và một class khi hiện thực một interface được gọi là can-do (hoặc –able).***

|  |  |
| --- | --- |
| **Abstract class** | **Interface** |
| Lớp trừu tượng có thể có các phương thức trừu tượng và không trừu tượng | Chỉ có phương thức trừu tượng |
| Không hỗ trợ đa kế thừa | Hỗ trợ đa kế thừa |
| Có thể có các biến final, non-final, static và non-static | Chỉ có các biến static và final |
| Có thể implement interface | Không thể extents lớp abstract |
| Từ khóa abstract được sử dụng để khai báo lớp trừu tượng | Từ khóa interface được sử dụng để khai báo giao diện |
| Một lớp trừu tượng có thể extents một lớp Java khác và thực hiện nhiều interface | Một interface chỉ có thể implement interface khác |
| Có thể được kế thừa bằng cách sử dụng từ khóa “extends” | Các lớp triển khai interface sử dụng từ khóa implements |
| Một lớp trừu tượng có thể có phạm vi truy cập là private, protected,… | Các phương thức của interface được mặc định là public |
| public abstract class Shape {        public abstract void draw();  } | public interface Drawable {  void draw();  } |
| Có thể xác định modifier | Mọi phương thức, property đều mặc định là public. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |