1. Static

* Từ khoá static được sử dụng chính để quản lý bộ nhớ
* Có thể áp dụng từ khoá static với các biến, phương thức, khối
* Từ khoá static thuộc về lớp chứ không thuộc về instance (thể hiện) của lớp
* Biến static: biến tĩnh, được sử dụng để tham chiếu thuộc tính chung của tất cả đối tượng, sử dụng biến static giúp chương trình sử dụng bộ nhớ hiêu quả hơn
* Phương thức static: 1 phương thức static thuộc lớp chứ không phải đối tượng của lớp, phương thức static có thể gọi mà không cần phải khởi tạo đối tượng, phương thức static có thể truy cập biến static và có thể thay đổi giá trị của nó, từ khoá this và super không thể sử dụng trong ngữ cảnh static, phương thức static không thể sử dụng biến non-static hoặc gọi trực tiếp phương thức non-static
* Khối static: được sử dụng để khởi tạo thành viên dữ liệu static, nó được thực thi trước main

1. Super và this

* Super được sử dụng để truy cập các phương thức của lớp cha, this được sử dụng để truy cập các phương thức của lớp hiện tại. Có thể sử dụng super và this ở bất cứ đâu, ngoại trừ trong static (không thể sử dụng trong khu vực static)
* Super: Từ khóa super trong Java là một biến tham chiếu được sử dụng để tham chiếu đối tượng lớp cha

Sử dụng: super có thể được sử dụng để tham chiếu đến biến cá thể lớp cha, super có thể được sử dụng để gọi phương thức lớp cha, super() có thể được sử dụng để gọi constructor lớp cha

* This: Từ khóa this trong Java là một biến tham chiếu được sử dụng để tham chiếu đối tượng lớp hiện tại

Sử dụng: this có thể được sử dụng để tham chiếu đến biến cá thể lớp hiện tại, this có thể được sử dụng để gọi phương thức lớp hiện tại, this() có thể được sử dụng để gọi constructor lớp hiện tại

1. Abstract class và interface

|  |  |
| --- | --- |
| Abstract class | Interface |
| Có thể có các phương thức abstract và non-abstract | Chỉ có thể có các phương thức abstract |
| Không hỗ trợ đa kế thừa | Hỗ trợ đa kế thừa |
| Có các biến final, non-final, static và non-static | Chỉ có các biến static và final |
| Có thể cung cấp việc triển khai, nội dung cài đặt cho phương thức của interface | Không thể cung cấp việc triển khai abstract class |
| Từ khoá abstract được sử dụng để khai báo abstract class | Từ khoá interface được sử dụng để khai báo interface |
| Một abstract class có thể kế thừa một class khác và implement nhiều interface | Một interface chỉ có thể kế thừa một interface khác |
| Một abstract class có thể được kế thừa bằng cách sử dụng từ khoá “extends” | Một interface có thể được implement bằng cách sử dụng từ khoá “implements” |
| Một lớp trừu tượng có thể có các thành viên lớp như private, protected… | Các thành viên của interface là public theo mặc định |

1. Enums

* Một enum là một kiểu dữ liệu đặc biệt được sử dụng để định nghĩa tập hợp các hằng số (các biến không thể thay đổi, như các biến final)
* Một enum có thể giống như một class, có các thuộc tính và phương thức. Sự khác biệt duy nhất là hằng số enum là public, static và final (không thể thay đổi - không thể bị ghi đè)
* Một enum không thể được sử dụng để tạo các đối tượng bởi vì nó chỉ có các constructor private và nó không thể kế thừa các lớp khác (nhưng nó có thể implement các interface)
* Sử dụng enums khi có các giá trị mà biết chắc rằng nó sẽ không thay đổi, như ngày tháng, các thứ trong tuần, các mùa trong năm, màu sắc…
* Phương thức values() trong enum trả về 1 mảng chứa tất cả các giá trị của enum