JAVA

Override (phải có extends): có nghĩa là chỉ thay đổi phần body của phương thức

Overloading (phải có extends):: có nghĩa là thay đổi phần tham số và kiểu dữ liệu trả về của phương thức

Abstract (Trừu tượng): nghĩa là ta sẽ khai báo 1 phương thức chung ở lớp cha. Sau đó các lớp con sẽ sử dụng phương thức đó nhưng sẽ hiệu chỉnh thêm (khá giống với Override). Lớp cha có những phương thức trừu tượng nào thì bắt buột lớp con cũng phải có những phương thức trừu tượng đó.

**Interface**: chứa các hành vi mà một class triển khai

**Functional interface**: là interface có duy nhất 1 abstract method

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Calculable func = new calculable();

Đầu tiên sẽ khai báo 1 lớp interface có phương thức trừu tượng tên là calculate

**Arrow Function:** chỉ có thể hoạt động trên funtional interface

**Enum:** (giống với class thông thường) nhưng sẽ khai báo trực tiếp giá trị

* Ngoài ra những giá trị này có thể chứa những thuộc tính, và những thuộc tính này cũg có thể chứa những phương thức như 1 class bth

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

**} Giá trị**

**Local:** Lấy đối tượng đại diện cho các ngôn ngữ và khu vực khác nhau trên toàn thế giới. VD:



**NumberFormat:** Định dạng kiểu số trong java. VD:



**Random:** Lấy số ngẫu nhiên. VD:



Cài đặt giá trị cho xucXac1:



**TimeUnit:** Đổi giờ sang giây, phút sang giờ, giờ sang ngày. VD:

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

**Output = 1h**



**Date:** Lấy thời gian. VD lấy thời gian hiện tại:



**DateFormat:** Định dạng ngày



**Calendar:** Lịch. VD:



**Stack: (Stack<obj> obj = new Stack<obj>();)**

Đối tượng vào sau ra trước, vào trước ra sau

* obj.push(o): đưa đối tượng vào Stack
* obj.pop(o): lấy đối tượng ra và xóa đối tượng khỏi Stack
* obj.peek(o): lấy đối tượng ra nhưng không xóa khỏi Stack
* obj.search(o): lấy vị trí của đối tượng trong stack
* obj.empty(): kiểm tra Stack có rỗng hay không

**Queue:** (Queue<obj> obj = new Queue<obj>();)

Đối tượng vào sau ra trước vào trước ra sau

Những phương thức của ***Queue***

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

**Set: (Set<Obj> obj = new Set<Obj>();)**

Là một collection không thể chứa 2 giá trị trùng lặp

Obj: kiểu dữ liệu hoặc đối tượng truyền vào

**Link:** https://viettuts.vn/java-collection/set-trong-java

* **HashSet:** Thêm vào mà không sắp xếp
* **TreeSet:** Thêm vào và có sắp xếp

**Map: (Map<Key, Value> obj = new Map<Key, Value>();)**

Key: khóa của map dùng để phân biệt giữa giá trị này và giá trị khác có trong map

Value: Giá trị của Key

* **HashMap:** Thêm phần tử vào và không sắp xếp
* **TreeMap:** Thêm phần tử vào và sắp xếp theo key của chúng

**Iterator:**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động**

**Geniric: (class Test<T>)**

**T:** là kiểu dữ liệu hoặc đối tượng truyền vào

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động