Java Language

1, Java Overview

* Is a popular programming language, created in 1995
* It’s owner by Oracle
* It’s used for : Mobie app, desk app, web app, web servers, games, database connection…
* Works on different platforms : Windows, Mac, Linux, Rapsberry Pi…
* Advantages :
  + Easy to learn and simple to use
  + Open-source and free
  + Secure, fast and powerful
  + Have a huge community support(Cộng đồng hỗ trợ lớn)

2,Constant, Variables and Operators

* Constant : value that cannot be changed after assigning(ko thay đổi sau khi gán).When declare,use static and final modifier, name convention is Upercase. Ex : static final int PRICE = 100;
* Variables : local variables, global variables, static variables
  + Local variables : A variable defined within 1 block or method or contructor is called local variable
  + Global variables : Scope is outside the function
  + Static variables : Declare similar as global variables, different that declared using static keyword.
  + Types : String(stores text), int(stores integer), float,double(real number), char(character), boolean(true,false)…
* Operators : performs operations on variable and values (contact with biến và giá trị)
  + Arithmetic Operator(toán tử số học) : + (Cộng), - (trừ), \* (nhân), / (chia) , % (chia dư), ++ (tăng giá trị lên 1), --(giảm gtri đi 1)
  + Comparision Operator(Toán tử so sánh): == (equal To), !=(Not equal to), > (greater than), < (Less than), >=(Greater than or equal to), <=(Less than or equal to)
  + Logical operator(toán tử logic) : && (logical and), ||(logical or), !(logical not)
  + Assignments Operators(toán tử gán) :

|  |  |  |
| --- | --- | --- |
| Operator | Ex | Same as |
| += | X += 5 | X = X+5 |
| -= | X -= 5 | X = X-5 |
| \*= | X\*=5 | X = X\*5 |
| /= | X /=5 | X = X/5 |
| %= | X %=5 | X = X%5 |
| = | X = 5 | X = 5 |

3. Control structure(Cấu trúc điều khiển)

* If-else : Cú pháp

If(condition1){

// block of code to be executed if the condition is true

}else if(condition2){

// block of code to be executed if the condition2 is true,1 false

}else{

// block of code to be executed if the condition1,2 is false

}

* SwitchCase : Cú pháp

switch(expression) {

case x:

// code block

break;

case y:

// code block

break;

default:

// code block

}

4. Loop

* While/Do While :
  + The while: loop loops through a block of code as long as a specified condition is true

While(Condition){ // code block to be executed }

* + Do while loop is child of while loop, excute do once, then check condition of while,if condition = true, it will repeat the loop

do {

// code block to be executed

}

while (condition);

* For/For each Loop :
  + for (statement 1; statement 2; statement 3) {

// code block to be executed

}

* + for (type variableName : arrayName) {

// code block to be executed

}

5, Arrays , String

* Arrays : store multiple values in a single variable
  + Declare : datatype [ ] name\_array ;
  + Size : name\_array = new datatype [size];
  + Truy suất : name\_array = [index];
  + Duyệt : For(int i =0;i < name\_array.length; i++){ name\_array[i]}
* String(Chuỗi) : class String, StringBuilder, StringTokenizer
  + Class String :

.Declare : String s1 = new String(“OK”); or String s1 = “OK”;

.Method : .length() : return size string

.charAt(int index) : return character at index

.trim() : remove khoảng trắng đầu cuối

.replace(char oldChar, char newChar): thay thế kí tự

.substring(int start,int end): return chuỗi từ start đến end

.compareTo(String S) : compare with s,return 0 nếu =, > 0 nếu s nhỏ hơn,<0 nếu s lớn hơn

.indexOf(String s) : find string s trong chuỗi, nếu tìm thấy return index s

.String pool : vùng nhớ nằm trong heap, lưu trữ các biến khai báo kiểu string

String a = “ok”; java truy cập vào tìm ô nhớ có cùng giá trị,ko có thì tự tạo ô nhớ trong String pool rồi tham chiếu.Khi khai báo kiểu đối tượng sẽ ko tìm mà lưu vào địa chỉ ô nhớ mới dù trùng giá trị trong heap => Tiết kiệm dung lượng heap

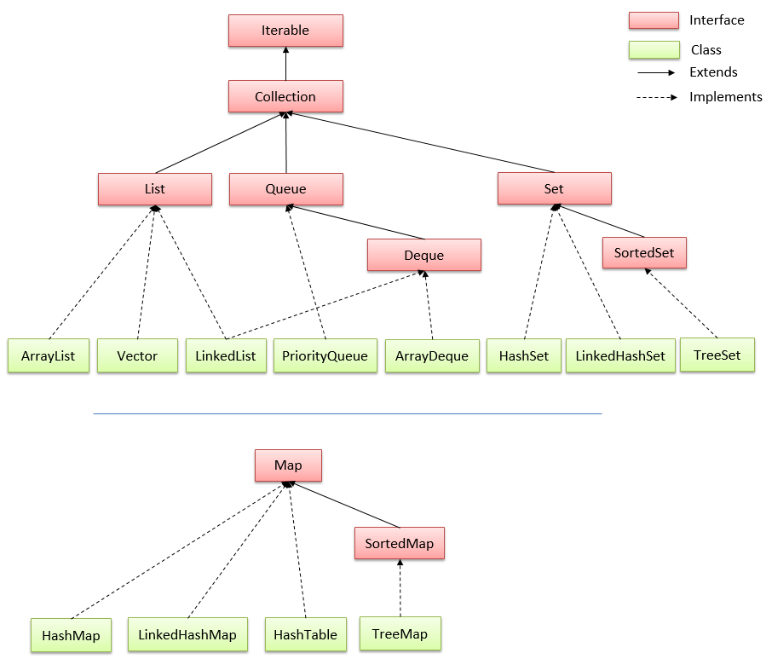
* + Class StringBuilder :

.Declare : StringBuilder a = new StringBuilder();

.Method : append() : add, insert() : chèn vào vị trí xác định, delete() : xóa, reverse() : đảo ngược chuỗi, toString() : Chuyển thành String

6. Collection

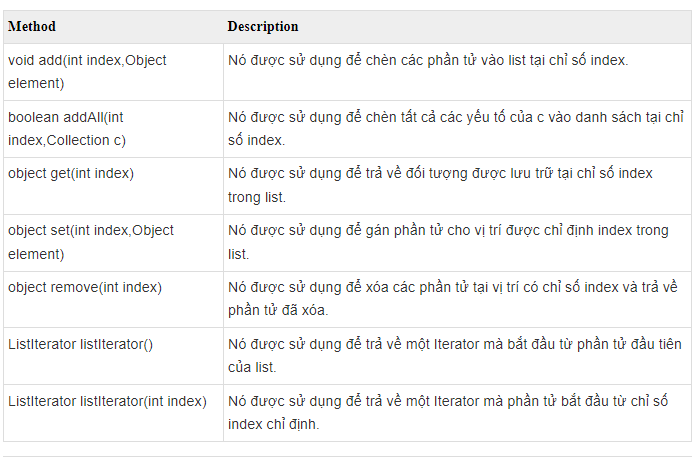
* Root Interface trong hệ thống cấp bậc collection. Cung cấp nhiều interface (Set,List,Queue…) và các class (arraylist,vector,linkedlist,PiorityQueue,HashSet,LinkHashSet,TreeSet)



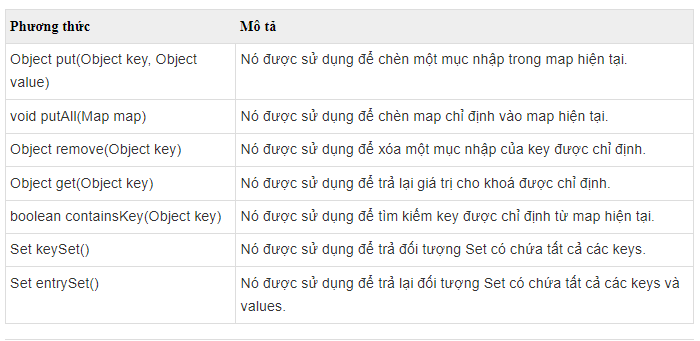
* + Set :
    - là 1 colletion không chứa 2 giá trị trùng lặp.Sử dụng để biểu diễn các bộ, ví dụ : Bộ tú lơ khơ,…
    - được triển khai bởi Hashset, LinkedHashset, Treeset, EsumSet
    - declare : Set<String> a = new HashSet<String>();
    - Method



* + List :
    - Chứa các phương thức để chèn và xóa phần tử dựa trên chỉ số index.
    - List<String> list = new Arraylist<String>();
    - Method :



* + Map :
    - Chứa cặp key-value, mỗi cặp key-value đc gọi là 1 entry, chỉ chứa các giá trị key duy nhất
    - Declare :
      * Sử dụng generic: Map<Integer, String> map = new HashMap<Integer,String>();
      * Ko sử dụng generic: Map map = new HashMap();
    - Method :



7.Exception

* Là một cơ chế xử lí các lỗi runtime : ClassNotFound, IO, SQL, Remote…..
* Gồm 2 loại là checked và unchecked :
  + Checked : Classes extends từ class Throwable ngoại trừ RuntimeException,Error gọi là checked exception. Các Checked Exception được kiểm tra tại compiler-time
  + Unchecked : Classes extends từ class RuntimeException được gọi là Unchecked exception