

Arraylist - Mảng động trong Java

* Đối với một số bài toán yêu cầu xử lý độc lập từng phần tử trong mảng và mảng không có kích thước cho trước thì ta có thể dùng hàm **scanner.hasNext()** để đọc liên tục các giá trị được nhập vào và xử lý chúng.

* Lưu ý: Hàm **scanner.hasNext()** chỉ sử dụng được đối với việc đọc xuất trên các trang nẹp bài tập, đọc xuất file. Hàm không chạy được trên dữ liệu nhập vào từ bàn phím người dùng.

❖ Để giải quyết các bài toán yêu cầu xử lý nhiều phần tử (hoặc đơn giản là ta cần một mảng mà không cần khai báo trước kích thước), ta có thể dùng ArrayList.

❖ Điểm nét về ArrayList:

1. Giới thiệu

- ArrayList được coi như mảng động trong Java
- ArrayList chỉ có thể lưu trữ đối tượng nên kiểu dữ liệu dùng khi khai báo ArrayList phải thuộc lớp Wrapper (Integer, Long, Float,...)

2. Khai báo

Có hai cách để khai báo ArrayList

```
List<data type> a = new ArrayList<>();
```

```
ArrayList<data type> a = new ArrayList<>();
```

3. Một số hàm thường dùng

a.add(value) [void] - Thêm giá trị *value* vào cuối mảng

a.add(index, value) [boolean] - Chèn giá trị *value* vào vị trí *index* của mảng

a.set(index, value) [int] - Thay đổi giá trị của phần tử có vị trí *index* thành *value*

a.get(index) [data type] - Trả về giá trị của phần tử có chỉ số là *index*

a.clear() [void] - Xóa toàn bộ phần tử của mảng

a.isEmpty() [boolean] - Kiểm tra mảng có rỗng (không chứa giá trị) hay không

a.contains(value) [boolean] - Kiểm tra xem giá trị *value* có xuất hiện trong mảng không

a.remove(index) [int] - Xóa phần tử có chỉ số là *index*

a.remove((Object)(value)) [boolean] - Xóa phần tử có chỉ số thấp nhất có giá trị là *value*

a.toArray() [object] - Trả về toàn bộ mảng (Dùng để sao chép mảng)

a.indexOf(value) [*int*] - Trả về chỉ số nhỏ nhất của phần tử có giá trị *value*, nếu không tồn tại giá trị *value* thì trả về **-1**

a.forEach((variable) -> *function*) [*void*] - Lấy từng phần tử của mảng làm tham số cho *function*