

LAB 5: ARRAYLIST

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Sử dụng ArrayList để duy trì tập hợp các phần tử
- ✓ Sử dụng thư viện tiện ích xử lý tập hợp Collections

Bài 1

Nhập danh sách số thực với số lượng tùy ý từ bàn phím sau đó xuất danh sách vừa nhập và tổng của nó.

HƯỚNG DẪN:

- ✓ Sử dụng ArrayList<Double> để duy trì danh sách số thực được nhập từ bàn phím
- ✓ Sử dụng vòng lặp while để nhập số lượng tùy ý

```
while(true){
    Double x = scanner.nextDouble();
    list.add(x);

    System.out.print("Nhập thêm (Y/N)? ");
    if(scanner.nextLine().equals("N")){
        break;
    }
}
```
- ✓ Sử dụng vòng lặp for-each để duyệt và xuất các phần tử của list ra màn hình

Bài 2

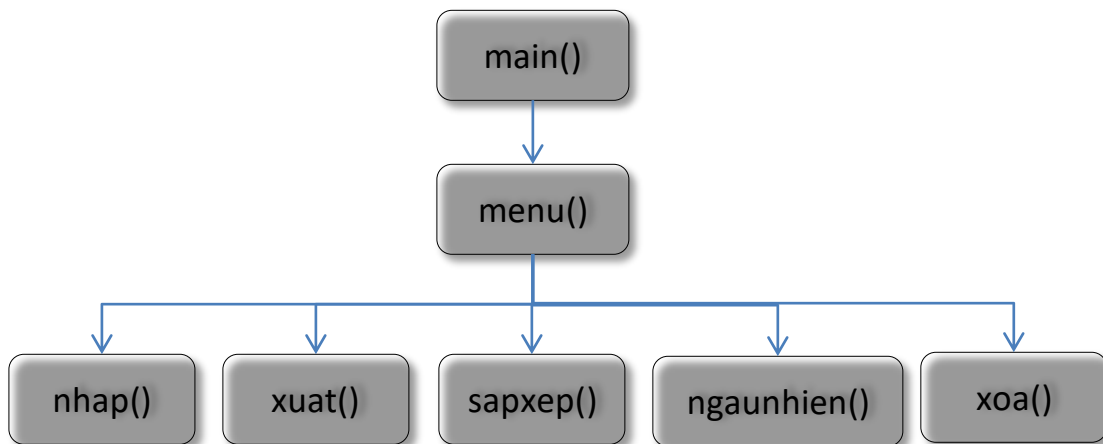
Viết chương trình thực hiện các chức năng sau

1. Nhập danh sách họ và tên
2. Xuất danh sách vừa nhập

3. Xuất danh sách ngẫu nhiên
4. Sắp xếp giảm dần và xuất danh sách
5. Tìm và xóa họ tên nhập từ bàn phím
6. Kết thúc

HƯỚNG DẪN

- ✓ Tổ chức ứng dụng theo sơ đồ sau



- ✓ Nhập và xuất tương tự bài 1
- ✓ Sử dụng hàm `Collections.shuffle(list)` để hoán đổi ngẫu nhiên các phần tử trong list
- ✓ Sử dụng `Collections.sort(list)` để sắp xếp tăng dần sau đó sử dụng `Collections.reverse(list)` để đảo các phần tử trong ArrayList
- ✓ Duyệt list và sử dụng `list.remove()` để xóa phần tử, dùng `break` để ngắt vòng lặp sau khi xóa

Bài 3

Xây dựng ứng dụng quản lý sản phẩm (thông tin mỗi sản phẩm gồm tên và giá) theo menu sau

1. Nhập danh sách sản phẩm từ bàn phím
2. Sắp xếp giảm dần theo giá và xuất ra màn hình
3. Tìm và xóa sản phẩm theo tên nhập từ bàn phím

4. Xuất giá trung bình của các sản phẩm

HƯỚNG DẪN

- ✓ Tổ chức ứng dụng tương tự bài 2
- ✓ Sử dụng `Collections.sort(list, comparator)` để sắp xếp danh sách sản phẩm với tiêu chí sắp xếp được định nghĩa như sau

```
Comparator<SanPham> comp = new Comparator<SanPham>() {  
    @Override  
    public int compare(SanPham o1, SanPham o2) {  
        return o1.donGia.compareTo(o2.donGia);  
    }  
};
```