

Bài 1.

Xây dựng đối tượng Student như sau:

- Thuộc tính: **id, name, score**.
- Phương thức:
 - **Phương thức khởi tạo 3 tham số** để xác định giá trị cho 3 thuộc tính
 - Phương thức khởi tạo từ chuỗi (id;name;score).
 - Override phương thức có sẵn là **toString** để trả về thông tin của đối tượng đó theo định dạng sau: [id_name_score_xep loai]
 - Các phương thức **get/set** cho các thuộc tính.
 - Phương thức nhập dữ liệu sinh viên từ bàn phím
 - Phương thức xếp loại sinh viên
 - Nếu điểm > 8: Loại giỏi
 - $6 < \text{Điểm} \leq 8$: Loại khá
 - Điểm ≤ 6 : Trung bình

Xây dựng đối tượng StudentManager như sau:

- Thuộc tính: **Student[], ArrayList, Vector, Hashtable: listStudents**.
- Phương thức:
 - **Phương thức khởi tạo**: tạo mới mảng học sinh với **5** phần tử. Và khởi tạo nên sẵn từng phần tử trong mảng với dữ liệu tùy ý.
 - Phương thức: **printList()**: in ra thông tin 5 học sinh.
 - Phương thức nhập danh sách sinh viên từ bàn phím
 - Phương thức: **sortByScoreDesc()**: sắp xếp 5 học sinh theo điểm giảm dần.
 - Sử dụng phương pháp đổi chỗ
 - Sử dụng interface Comparable
 - Sử dụng interface Comparator
 - Phương thức: **sortByIDDesc()**: sắp xếp 5 học sinh theo ID giảm dần.
 - Sử dụng phương pháp đổi chỗ
 - Sử dụng interface Comparable
 - Sử dụng interface Comparator
 - Phương thức: **findStudentByScore(double score)**: trả về học sinh có điểm bằng với tham số được truyền vào. Nếu không có thì trả về giá trị **null**.
 - Phương thức: **findStudentGood()**: Trả về sinh viên loại giỏi
 - Phương thức Kiểm tra sự tồn tại của sinh viên: sinh viên gọi là đã tồn tại nếu trong danh sách đã chứa ID của sinh viên vừa nhập
 - **Xây dựng phương thức ghi dữ liệu từ listStudents vào file**
 - **Xây dựng phương thức đọc dữ liệu từ file vào listStudents**

Xây dựng đối tượng Main và chạy thử các đối tượng.

Bài 2.

Xây dựng class Danh Bạ bao gồm: Tên, Số ĐT có kiểu String

1. Xây dựng phương thức khởi tạo: không tham số, có tham số
2. Phương thức khởi tạo từ chuỗi, các thành phần cách nhau dấu ;
3. Phương thức nhập danh bạ từ bàn phím
4. Phương thức toString

Xây dựng class Quản lý Danh Bạ như sau:

1. Xây dựng lớp Danh Bạ, gồm phương thức khởi tạo để nhập dữ liệu cho Danh Bạ, phương thức getTen, getSDT
2. Xây dựng lớp Quản lý gồm:
 - a. **Vector, ArrayList:** ListDB, lưu ý mỗi danh bạ số điện thoại và tên đều phải là duy nhất, không được trùng nhau
 - b. PT Khởi tạo, tạo danh sách với 5 đối tượng danh bạ
 - c. PT Kiểm Tra Tồn Tại (String SDT). Danh bạ được gọi là trùng nhau khi số điện thoại giống nhau
 - i. Nếu tồn tại DB thì trả về true, ngược lại trả về false
 - d. PT in danh bạ không tham số, dùng để in listDB
 - e. PT in danh bạ có 1 tham số, dùng để in danh sách được truyền vào
 - f. PT nhập danh sách danh bạ từ bàn phím
 - g. PT thêm danh bạ(String hoTen, String sdt) để thêm dữ liệu vào listDB
 - h. PT sửa danh bạ(String sdtcu, String SDTmới) để sửa lại sdt với tên được nhập vào, nếu sửa thành công trả về true, ngược lại thì trả về false
 - i. PT sắp xếp danh bạ theo họ tên theo thứ tự a-z, nếu 2 người có họ và tên trùng nhau thì sắp xếp theo số điện thoại.
 - i. Sắp xếp danh bạ theo cách đổi chỗ
 - ii. Sắp xếp danh bạ sử dụng interface Comparable
 - iii. Sắp xếp danh bạ sử dụng interface Comparator
 - j. PT Tìm kiếm DB(String ten) như sau:
 - i. Tìm kiếm tất cả các DB có chứa tên trùng với tên đã nhập
 - k. **Xây dựng phương thức ghi dữ liệu từ ListDB vào file**
 - l. **Xây dựng phương thức đọc dữ liệu từ file vào ListDB**

Xây dựng class Main để test các phương thức

Câu 2:

Class DanhBa.java

```
public class DanhBa {
    private String hoTen;
    private String soDienThoai;

    public DanhBa() {
    }

    public DanhBa(String hoTen, String soDienThoai) {
        this.hoTen = hoTen;
        this.soDienThoai = soDienThoai;
    }
    //hoten;sodienthoai
    public DanhBa(String s) {
        String[] tokens = s.split(";");
        hoTen = tokens[0];
        soDienThoai = tokens[1];
    }

    public void nhapDanhBa() {
        Scanner nhap = new Scanner(System.in);
        System.out.println("Nhap Ho ten: ");
        hoTen = nhap.nextLine();
        System.out.println("Nhap So dien thoai: ");
        soDienThoai = nhap.nextLine();
    }

    @Override
    public String toString() {
        return hoTen+"; " + soDienThoai;
    }

    public String getSoDienThoai() {
        return soDienThoai;
    }

    public void setSoDienThoai(String soDienThoai) {
        this.soDienThoai = soDienThoai;
    }

    public String getHoTen() {
        return hoTen;
    }
}
```

Class QuanLyDanhBa.java

Code minh họa sử dụng ArrayList.

Thay ArrayList bằng Vector được cách cài đặt sử dụng Vector

```
Vector<DanhBa> listDB;  
listDB = new Vector<>();  
public Vector<DanhBa> timKiemTen(String ten)  
public void inDanhBa(Vector<DanhBa> list)  
public Vector<DanhBa> timKiemTen(String ten)
```

```
public class QuanLyDanhBa {  
    private ArrayList<DanhBa> listDB;  
  
    public QuanLyDanhBa() {  
        listDB = new ArrayList<>();  
        DanhBa db0 = new DanhBa("Le Van A", "0988765435");  
        DanhBa db1 = new DanhBa("Le Van A", "094576435");  
        DanhBa db2 = new DanhBa("Phung Van K", "093465435");  
        DanhBa db3 = new DanhBa("Pham thi H", "097975435");  
        DanhBa db4 = new DanhBa("Nguyen Van O", "093465435");  
  
        listDB.add(db0);  
        listDB.add(db1);  
        listDB.add(db2);  
        listDB.add(db3);  
        listDB.add(db4);  
    }  
  
    public boolean kiemTrDanhBa(String sdt) {  
        for (int i = 0; i < listDB.size(); i++) {  
            if (listDB.get(i).getSoDienThoai().equals(sdt)) {  
                return true;  
            }  
        }  
        return false;  
    }  
  
    public void inDanhBa() {  
        for (int i = 0; i < listDB.size(); i++) {  
            System.out.println(listDB.get(i).toString());  
        }  
    }  
  
    public void inDanhBa(ArrayList<DanhBa> list) {  
        if (list.isEmpty()) {  
            System.out.println("Khong to tai danh sach danh ba");  
        } else {  
            for (int i = 0; i < list.size(); i++) {  
                System.out.println(list.get(i).toString());  
            }  
        }  
    }  
  
    public void themDanhBa(String hoTen, String sdt) {
```

```

        DanhBa db = new DanhBa(hoTen, sdt);
        listDB.add(db);
    }

    public void nhapDanhBa() {
        for (int i = 0; i < 2; i++) {
            DanhBa db = new DanhBa();
            db.nhapDanhBa();
            listDB.add(db);
        }
    }

    public boolean suaDanhBa(String sdtCu, String sdtMoi){
        for (int i = 0; i < listDB.size(); i++) {
            if(listDB.get(i).getSoDienThoai().equals(sdtCu)) {
                //listDB.get(i).setSoDienThoai(sdtMoi);
                DanhBa db = new DanhBa(listDB.get(i).getHoTen(), sdtMoi);
                listDB.set(i, db);
                return true;
            }
        }
        return false;
    }
}

```

Sắp xếp sử dụng cách đổi chỗ, tìm hiểu cách sắp xếp sử dụng interface Comparable hoặc interface Comparator.

[Link đọc tìm hiểu](#) (Giữ control để mở Link)

```

public void sapXepHoTen(){
    DanhBa db;
    for (int i = 0; i < listDB.size(); i++) {
        for (int j = i+1; j < listDB.size(); j++) {
            if(listDB.get(i).getHoTen().compareTo(listDB.get(j).getHoTen())>0) {
                db = listDB.get(i);
                listDB.set(i, listDB.get(j));
                listDB.set(j, db);
            } else if(listDB.get(i).getHoTen().compareTo(listDB.get(j).getHoTen())==0) {
                if(listDB.get(i).getSoDienThoai().compareTo(listDB.get(j).getSoDienThoai())>0) {
                    db = listDB.get(i);
                    listDB.set(i, listDB.get(j));
                    listDB.set(j, db);
                }
            }
        }
    }
}

```

```

public ArrayList<DanhBa> timKiemTen(String ten){
    ArrayList<DanhBa> listTimKiem = new ArrayList<>();
    for (int i = 0; i < listDB.size(); i++) {
        if(listDB.get(i).getHoTen().equals(ten)) {
            listTimKiem.add(listDB.get(i));
        }
    }
    return listTimKiem;
}
}

```

Class Main.java

```
public class Main {  
    public static void main(String[] args) {  
        QuanLyDanhBa qldb = new QuanLyDanhBa();  
        System.out.println("=====IN DANH BA KHOI TAO=====");  
        qldb.inDanhBa();  
        System.out.println("===SUA DANH BA ===");  
        qldb.suaDanhBa("0988765435", "0000765435");  
        qldb.inDanhBa();  
        System.out.println("===NHAP DANH BA ===");  
        qldb.nhapDanhBa();  
        System.out.println("=====IN DANH BA VUA NHAP =====");  
        qldb.inDanhBa();  
        System.out.println("===SAP XEP DANH BA THEO TEN ===");  
        qldb.sapXepHoTen();  
        qldb.inDanhBa();  
        System.out.println("=====TIM KIEM DANH BA ===");  
        qldb.inDanhBa(qldb.timKiemTen("Le Van A"));  
    }  
}
```