BÁO CÁO BÀI KIỂM TRA SỐ 3

**Họ và tên:** Nguyễn Hoàng Long

**Mã SV:** 2018602388KHMT1

*1) Chọn một trong 2 định dạng file (văn bản hoặc nhị phân)*

* Sinh ngẫu nhiên một danh sách nhân viên (số lượng từ 50-100 nhân viên)
* Sắp xếp theo tên hoặc/và tuổi.
* Ghi vào file với định dạng file xác định ở trên.

*2) Xây dựng một interface Manager* a) Các phương thức

* public boolean addEmployee(Person p);
* public boolean editEmployee(Person p);
* public boolean delEmployee(Person p);
* public ArrayList<Employee> searchEmployee(Person[] list, String name);
* public ArrayList<Employee> searchEmployee(Person[] list, Address addr);

b) Cập nhật với file

* public ArrayList<Employee> addEmployee(File f);
* public ArrayList<Employee> editEmployee(File f);
* public ArrayList<Employee> delEmployee(File f);

**Câu 1:**

* Sinh ngẫu nhiên một danh sách nhân viên

1. Khởi tạo lớp Person với các thuộc tính và phương thức cần thiết
2. Khởi tạo lớp Employee kế thừa lớp Person

**public** **class** Employee **extends** Person **implements** Comparable<Employee>

{

**private** **int** net;

**private** String position;

**private** **short** apply\_year;

**private** **short** depart\_id;

**public** Employee(String firstname, String lastname, **int** age, Address address, **int** net, String position, **short** apply\_year, **short** depart\_id)

{

**super**(firstname, lastname, age, address);

**this**.net = net;

**this**.position = position;

**this**.apply\_year = apply\_year;

**this**.depart\_id = depart\_id;

}

@Override

**public** String toString()

{

**return** **super**.toString()+" Net=" + net + " Position=" + position + " Apply year=" + apply\_year + " Depart id=" + depart\_id ;

}

}

1. Sinh ngẫu nhiên danh sách Employee thông qua phương thức

Employee[] generateEmployee(**int** n)

//Chạy phương thức generateEmployee() trong hàm main

**public** **static** **void** main(String[] args) **throws** IOException {

Employee[] list = *generateEmployee*(4);

List<Employee> l = **new** ArrayList<>();

Collections.*addAll*(l, list);

System.***out***.println("Danh sách Employee vừa khởi tạo");

*printEmployee*(l);

*}*

**Danh sách Employee vừa khởi tạo**

1 FirstName=Long LastName=Khuất Age=49 Address=Bùi\_Viện,Long\_Biên,Hà\_Nội Net=0 Position=null Apply year=0 Depart id=0

2 FirstName=Cường LastName=Nguyễn Age=73 Address=Hàng\_mã,Từ\_Liêm,Hà\_Nội Net=0 Position=null Apply year=0 Depart id=0

3 FirstName=Thanh LastName=Nguyễn Age=3 Address=Hàng\_mã,Từ\_Liêm,Hà\_Nội Net=0 Position=null Apply year=0 Depart id=0

4 FirstName=Anh LastName=Hoàng Age=73 Address=Hàng\_mã,Long\_Biên,Hà\_Nội Net=0 Position=null Apply year=0 Depart id=0

* Sắp xếp theo tên hoặc/và tuổi.

1. Khi có danh sách Employee ta tiến hành sắp xếp sử dụng Interface **Comparable** bằng cách implements Interface **Comparable** để cài đặt **phương thức compareTo()**.

Ta cài đặt tiêu chí để so sánh trong phương thức **compareTo()** để so sánh các đối tượng Employee với nhau theo **age**

@Override

**public** **int** compareTo(Employee o) {

**return** **this**.getAge() - o.getAge();

}

Phương thức **Collections.sort**(ArrayList<Employee> tmp) được sử dụng để sắp xếp các phần tử của list Employee theo age

**public** **static** ArrayList<Employee> sortByAge(Employee[] list){

ArrayList<Employee> tmp = **new** ArrayList<>();

Collections.*addAll*(tmp, list);

Collections.*sort*(tmp);

**return** tmp;

}

**Danh sách Employee sau khi sắp xếp theo tuổi tăng dần**

1 FirstName=Thanh LastName=Nguyễn Age=3 Address=Hàng\_mã,Từ\_Liêm,Hà\_Nội Net=0 Position=null Apply year=0 Depart id=0

2 FirstName=Long LastName=Khuất Age=49 Address=Bùi\_Viện,Long\_Biên,Hà\_Nội Net=0 Position=null Apply year=0 Depart id=0

3 FirstName=Cường LastName=Nguyễn Age=73 Address=Hàng\_mã,Từ\_Liêm,Hà\_Nội Net=0 Position=null Apply year=0 Depart id=0

4 FirstName=Anh LastName=Hoàng Age=73 Address=Hàng\_mã,Long\_Biên,Hà\_Nội Net=0 Position=null Apply year=0 Depart id=0

1. Sắp xếp sử dụng Interface **Comparator và** tạo đối tượng nặc danh **Comparator** để cài đặt **phương thức compare()**. Cài đặt tiêu chí để so sánh trong phương thức **compare()** để so sánh các đối tượng Employee với nhau theo **firstname** và **lastname**

Phương thức **Collections.sort**(ArrayList<Employee> tmp, Comparator<Emloyee> e) được sử dụng để sắp xếp các phần tử của list Employee theo age

**public** **static** ArrayList<Employee> sortByName(Employee[] list)

{

//khai báo mảng kết quả

ArrayList<Employee> tmp = **new** ArrayList<>();

Collections.*addAll*(tmp, list);

Collections.*sort*(tmp, **new** Comparator<Employee>() {

@Override

**public** **int** compare(Employee o1, Employee o2)

{

**int** resFirst = o1.getFirstName().compareTo(o2.getFirstName());

**if** (resFirst != 0)

{

**return** resFirst;

} **else**

{

**int** resLast = o1.getLastName().compareTo(o2.getLastName());

**return** resLast;

}

}

});

**return** tmp;

}

**Danh sách Employee sau khi sắp xếp theo tên theo thứ tự alphabet**

Danh sách Employee sau khi sắp xếp tên

1 FirstName=Anh LastName=Hoàng Age=73 Address=Hàng\_mã,Long\_Biên,Hà\_Nội Net=0 Position=null Apply year=0 Depart id=0

2 FirstName=Cường LastName=Nguyễn Age=73 Address=Hàng\_mã,Từ\_Liêm,Hà\_Nội Net=0 Position=null Apply year=0 Depart id=0

3 FirstName=Long LastName=Khuất Age=49 Address=Bùi\_Viện,Long\_Biên,Hà\_Nội Net=0 Position=null Apply year=0 Depart id=0

4 FirstName=Thanh LastName=Nguyễn Age=3 Address=Hàng\_mã,Từ\_Liêm,Hà\_Nội Net=0 Position=null Apply year=0 Depart id=0

* Ghi vào file với định dạng file xác định

1. Sử dụng **lớp FileWriter** để ghi các dữ liệu theo định dạng ký tự vào một file.

//xác định tệp tin lưu trữ dữ liệu bằng đường dẫn

FileWriter outFileWriter = **new** FileWriter(filepath);

Sử dụng **lớp PrintWriter** để ghi các định dạng đại diện của các đối tượng vào stream hướng văn bản.

//tạo đối tượng thực thi ghi dữ liệu

PrintWriter out = **new** PrintWriter(outFileWriter);

1. Phương thức **textOutputFile**(String filepath, ArrayList<Employee> list)

//Phương thức ghi dữ liệu ra file có đường dẫn truyền vào

**public** **static** **void** textOutputFile(String filepath, ArrayList<Employee> list)

{

//xác định tệp tin lưu trữ dữ liệu

FileWriter outFileWriter = **new** FileWriter(filepath);

//tạo đối tượng thực thi ghi dữ liệu

PrintWriter out = **new** PrintWriter(outFileWriter);

//ghi tiêu đề ra file

out.println("\t\tDANH SACH EMPLOYEE");

out.println(String.*format*("%-10s%-10s%-10s%-10s%-10s%-45s%-10s%-10s%-10s%-10s","STT" ,"ID", "FirstName", "LastName", "Age", "Address",

"Net", "Position", "ApplyYear", "DepartId"));

out.println("--------------------------------------------------------");

**for**(**int** i=0;i<list.size();i++)

{

//Định dạng và ghi dữ liệu ghi ra file

out.println(String.*format*("%-10d%-10d%-10s%-10s%-10d%-45s%-10d%-10s%-10d%-10d",i+1, list.get(i).getId() ,list.get(i).getFirstName(), list.get(i).getLastName(), list.get(i).getAge(), list.get(i).getAddress(),

list.get(i).getNet(), list.get(i).getPosition(), list.get(i).getApply\_year(), list.get(i).getDepart\_id()));

}

//Đóng luồng

out.close();

}

Chạy phương thức textOutputFile("danhsach.txt", list) ghi kết quả ra file **danhsach.txt**

DANH SACH EMPLOYEE

STT FirstName LastName Age Address Net Position

------------------------------------------------------------

1 Anh Hoàng 73 Hàng\_mã,Long\_Biên,Hà\_Nội 0 null

2 Cường Nguyễn 73 Hàng\_mã,Từ\_Liêm,Hà\_Nội 0 null

3 Long Khuất 49 Bùi\_Viện,Long\_Biên,Hà\_Nội 0 null

4 Thanh Nguyễn 3 Hàng\_mã,Từ\_Liêm,Hà\_Nội 0 null

1. Phương thức **textInputFile**(String filepath)

//đọc dữ liệu từ file có đường dẫn truyền vào

**public** **static** ArrayList<Employee> textInputFile(String filepath)

{

ArrayList<Employee> list=**null**;

//xác định tệp tin lưu trữ dữ liệu

FileReader inFileReader = **new** FileReader(filepath);

//Khai báo bộ đệm đọc

BufferedReader in = **new** BufferedReader(inFileReader);

//khởi tạo danh sách dữ liệu trả về

list = **new** ArrayList<Employee>();

//Bỏ qua 3 dòng đầu không đọc

String l1 = in.readLine();

String l2 = in.readLine();

String l3 = in.readLine();

//Đếm số dòng trong danh sách file

Path file = Paths.*get*(filepath);

**long** count = Files.*lines*(file).count();

//Cắt dòng đọc được thành 1 mảng

**for**(**int** i=0;i<count-3;i++)

{

String line = in.readLine();

**while**(line.indexOf(" ") != -1)

{

// Thay 2 cách = 1 dấu cách trong dòng đọc được

line = line.replace(" ", " ");

}

//Cắt dòng đọc được thành 1 mảng bằng dấu cách

String []emp = line.split(" ");

//Lưu phần tử thứ 5, 6, 7 vào Address

Address a = **new** Address(emp[7], emp[6], emp[5]);

//Nạp từng phần tử của mảng vào employee

Employee employee = **new** Employee(Integer.*parseInt*(emp[1]), emp[2], emp[3], Integer.*parseInt*(emp[4]), a , Integer.*parseInt*(emp[8]), emp[9], Short.*parseShort*(emp[10]), Short.*parseShort*(emp[11]));

list.add(employee);

}

//Đóng luồng

in.close();

**return** list;

}

Chạy phương thức textInputFile("danhsach.txt") trả về kết quả ra console

1 Anh Hoàng 73 Hàng\_mã,Long\_Biên,Hà\_Nội 0 null

2 Cường Nguyễn 73 Hàng\_mã,Từ\_Liêm,Hà\_Nội 0 null

3 Long Khuất 49 Bùi\_Viện,Long\_Biên,Hà\_Nội 0 null

4 Thanh Nguyễn 3 Hàng\_mã,Từ\_Liêm,Hà\_Nội 0 null

**Câu 2 :**

* Xây dựng interface **Manager** với các phương thức

**public** **interface** Manager

{

**public** **boolean** addEmployee(Person p);

**public** **boolean** editEmployee(Person p);

**public** **boolean** delEmployee(Person p);

**public** ArrayList<Employee> searchEmployee(Employee[] list, String name);

**public** ArrayList<Employee> searchEmployee(Employee[] list, Address addr);

}

1. Xây dựng interface **FileManager** kế thừa **Manager** với các phương thức cập nhật với file

**public** **interface** FileManager **extends** Manager

{

**public** ArrayList<Employee> addEmployee(File f);

**public** ArrayList<Employee> editEmployee(File f);

**public** ArrayList<Employee> delEmployee(File f);

}

1. Xây dựng class **TestManager** triển khai **interface FileManager** ghi đè các phương thức từ **FileManager** và **Manager**

**public** **class** TestManager **implements** FileManager

{

**@override**

**public** **boolean** addEmployee(Person p);

**@override**

**public** **boolean** editEmployee(Person p);

**@override**

**public** **boolean** delEmployee(Person p);

**@override**

**public** ArrayList<Employee> searchEmployee(Employee[] list, String name);

**@override**

**public** ArrayList<Employee> searchEmployee(Employee[] list, Address addr);

**@override**

**public** ArrayList<Employee> addEmployee(File f);

**@override**

**public** ArrayList<Employee> editEmployee(File f);

**@override**

**public** ArrayList<Employee> delEmployee(File f);

}

* Các phương thức

Khởi tạo 1 danh sách nhân viên rồi ghi ra file danhsach.txt

DANH SACH EMPLOYEE

STT ID FirstName LastName Age Address Net Position ApplyYear DepartId

-----------------------------------------------------------------

1 0 An Chu 70 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

2 0 Anh Nguyễn 11 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

3 0 Anh Vũ 1 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

4 0 Cường Võ 36 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

1. Phương thức thêm nhân viên vào file

//phương thức thêm nhân viên vào file có đường dẫn filepath

@Override

**public** **boolean** addEmployee(Person p, String filepath)

{

//Khởi tạo danh sách nhân viên

ArrayList<Employee> employees = **null**;

**if**(p==**null**) **return** **false**;

**try** {

// Đọc danh sách nhân viên từ file

employees = MyFile.*textInputFile*(filepath);

// Thêm nhân viên vào danh sách

employees.add((Employee) p);

// Cập nhật danh sách nhân viên trong file

MyFile.*textOutputFile*(filepath, employees);

**return** **true**;

} **catch** (IOException e) {

e.printStackTrace();

**return** **false**;

}

}

Chạy phương thức trong main, ta thấy nhân viên có id = 1 xuất hiện

Person p = **new** Employee(1,"Long", "Nguyễn", 20, **new** Address(), 0, **null**, (**short**)0, (**short**)0);

TestManager t = **new** TestManager();

**t.addEmployee**(p, "danhsach.txt");

DANH SACH EMPLOYEE

STT ID FirstName LastName Age Address Net Position ApplyYear DepartId

-----------------------------------------------------------------

1 0 An Chu 70 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

2 0 Anh Nguyễn 11 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

3 0 Anh Vũ 1 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

4 0 Cường Võ 36 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

5 1 Long Nguyễn 20 NoStreetName NoDistrictName NoCityName 0 null 0 0

1. Phương thức sửa nhân viên và cập nhật lại file

//phương thức sửa nhân viên

@Override

**public** **boolean** editEmployee(Person p, String filepath)

{

//Khởi tạo danh sách nhân viên

ArrayList<Employee> employees = **null**;

**if**(p==**null**) **return** **false**;

**try** {

// Đọc danh sách nhân viên từ file

employees = MyFile.*textInputFile*(filepath);

**for**(**int** i = 0;i<employees.size();i++)

{

// Kiểm tra xem mã nhân viên nhập vào có trong file hay không

**if**(employees.get(i).getId()==((Employee) p).getId())

{

employees.set(i, (Employee)p);

**break**;

}

}

// Cập nhật lại file

MyFile.*textOutputFile*(filepath, employees);

**return** **true**;

} **catch** (IOException e) {

e.printStackTrace();

**return** **false**;

}

}

Chạy phương thức trong main, thấy nhân viên có id=1 đã bị sửa lại

//update lại firstname, lastname, age address, net, position cho employee có id = 1

Person p = **new** Employee(1,"Phương", "Minh", 20, **new** Address(), 0, **null**, (**short**)0, (**short**)0);

TestManager t = **new** TestManager();

**t.addEmployee**(p, "danhsach.txt");

DANH SACH EMPLOYEE

STT ID FirstName LastName Age Address Net Position ApplyYear DepartId

-----------------------------------------------------------------

1 0 An Chu 70 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

2 0 Anh Nguyễn 11 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

3 0 Anh Vũ 1 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

4 0 Cường Võ 36 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

5 1 Phương Minh 25 NoStreetName NoDistrictName NoCityName 1 null 3 3

1. Phương thức xóa nhân viên trong file

//phương thức xóa nhân viên

@Override

**public** **boolean** delEmployee(Person p, String filepath)

{

//Khởi tạo danh sách nhân viên

ArrayList<Employee> employees = **null**;

**if**(p==**null**) **return** **false**;

**try** {

// Đọc danh sách nhân viên từ file

employees = MyFile.*textInputFile*(filepath);

**for**(**int** i = 0;i<employees.size();i++)

{

// Kiểm tra xem mã nhân viên nhập vào có trong file hay không

**if**(employees.get(i).getId()==((Employee) p).getId())

{

// xóa nhân viên khỏi danh sách

employees.remove(i);

**break**;

}

}

// Cập nhật lại file

MyFile.*textOutputFile*(filepath, employees);

**return** **true**;

} **catch** (IOException e) {

e.printStackTrace();

**return** **false**;

}

}

Chạy phương thức trong main ta thấy nhân viên có id = 1 đã bị xóa

Person p = **new** Employee(1,"Phương", "Minh", 20, **new** Address(), 0, **null**, (**short**)0, (**short**)0);

TestManager t = **new** TestManager();

**t.delEmployee**(p, "danhsach.txt");

DANH SACH EMPLOYEE

STT ID FirstName LastName Age Address Net Position ApplyYear DepartId

-----------------------------------------------------------------

1 0 An Chu 70 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

2 0 Anh Nguyễn 11 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

3 0 Anh Vũ 1 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

4 0 Cường Võ 36 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

1. Phương thức tìm nhân viên trong danh sách theo tên

//phương thức tìm nhân viên trong danh sách theo tên

@Override

**public** ArrayList<Employee> searchEmployee(Person[] list, String name)

{

//Khởi tạo danh sách nhân viên cần tìm

ArrayList<Employee> result = **new** ArrayList<Employee>();

**for**(Person p : list)

{ //Tìm nhân viên có tên nhập vào trong danh sách

**if**(p.getFirstName().equals(name))

{ //Thêm nhân viên vừa tìm vào danh sách nhân viên cần tìm

result.add((Employee) p);

}

}

**return** result;

}

Chạy phương thức trong main

//Tìm nhân viên có tên là “Anh”

MyEmployee.*printEmployee*(t.searchEmployee(array, "Anh"));

1 ID=0 FirstName=Anh LastName=Nguyễn Age=11 Address=NoStreetName NoDistrictName NoCityName Net=0 Position=NONE Apply year=2021 Depart id=0

2 ID=0 FirstName=Anh LastName=Vũ Age=1 Address=NoStreetName NoDistrictName NoCityName Net=0 Position=NONE Apply year=2021 Depart id=0

1. Phương thức tìm nhân viên trong danh sách theo địa chỉ

//phương thức tìm nhân viên theo địa chỉ

@Override

**public** ArrayList<Employee> searchEmployee(Person[] list, Address addr)

{

//Khởi tạo danh sách nhân viên

ArrayList<Employee> result = **new** ArrayList<Employee>();

**for**(Person p : list)

{

//Tìm nhân viên có address nhập vào trong danh sách

**if**(p.getAddress().equals(addr))

{ //Thêm nhân viên vừa tìm vào danh sách nhân viên cần tìm

result.add((Employee) p);

}

}

**return** result;

}

1. Phương thức nhập nhân viên sử dụng trong **addEmployee(File f)**

//phương thức nhập nhân viên mới

**public** **static** Employee nhap()

{ //khởi tạo nhân viên

Employee e = **new** Employee();

Scanner s = **new** Scanner(System.***in***);

System.***out***.println("Nhập nhân viên: ");

**boolean** check = **false**;

**do** {

**try** {

//Nhập mã nhân viên

System.***out***.print("Id: ");

e.setId(Integer.*parseInt*(s.nextLine()));

System.***out***.print("First name : ");

e.setFirstName(s.nextLine());

System.***out***.print("Last name : ");

e.setLastName(s.nextLine());

System.***out***.print("Age : ");

e.setAddress(**new** Address());

e.setAge(Integer.*parseInt*(s.nextLine()));

System.***out***.print("NET : ");

e.setNet(Integer.*parseInt*(s.nextLine()));

System.***out***.print("Position : ");

e.setPosition(s.nextLine());

System.***out***.print("Apply year : ");

e.setApply\_year(Short.*parseShort*(s.nextLine()));

System.***out***.print("Depart id : ");

e.setDepart\_id(Short.*parseShort*(s.nextLine()));

check = **true**;

}**catch** (Exception ex) {

System.***out***.println("Bạn nhập sai kiểu số");

check = **false**;

}

}**while**(!check);

System.***out***.println("Đã Thoát");

**return** e;

}

1. Phương thức sửa nhân viên có mã nhập vào

//phương thức sửa nhân viên dựa vào mã

**public** **static** Employee sua()

{ //khởi tạo nhân viên

Employee e = **new** Employee();

Scanner s = **new** Scanner(System.***in***);

System.***out***.println("Nhập nhân viên: ");

**boolean** check = **false**;

**do** {

**try** {

//Nhập mã nhân viên cần sửa

System.***out***.print("Nhập mã nhân viên cần sửa : ");

e.setId(Integer.*parseInt*(s.nextLine()));

System.***out***.print("Nhập thông tin nhân viên cần sửa : ");

System.***out***.print("First name : ");

e.setFirstName(s.nextLine());

System.***out***.print("Last name : ");

e.setLastName(s.nextLine());

System.***out***.print("Age : ");

e.setAddress(**new** Address());

e.setAge(Integer.*parseInt*(s.nextLine()));

System.***out***.print("NET : ");

e.setNet(Integer.*parseInt*(s.nextLine()));

System.***out***.print("Position : ");

e.setPosition(s.nextLine());

System.***out***.print("Apply year : ");

e.setApply\_year(Short.*parseShort*(s.nextLine()));

System.***out***.print("Depart id : ");

e.setDepart\_id(Short.*parseShort*(s.nextLine()));

check = **true**;

}**catch** (Exception ex) {

System.***out***.println("Nhập sai kiểu số");

check = **false**;

}

}**while**(!check);

System.***out***.println("Đã Thoát");

**return** e;

}

1. Phương thức thêm nhân viên vào file

//phương thức thêm nhân viên cập nhật với file

@Override

**public** ArrayList<Employee> addEmployee(File f)

{

//Khởi tạo danh sách nhân viên

ArrayList<Employee> employees = **null**;

//Nhập vào thông tin nhân viên

Employee employee = ***nhap*();**

**try** {

// Đọc danh sách nhân viên từ file

employees = MyFile.*textInputFile*(f.getPath());

//Thêm nhân viên vào danh sách nhân viên

employees.add(employee);

// Cập nhật lại file

MyFile.*textOutputFile*(f.getPath(), employees);

} **catch** (IOException e1) {

e1.printStackTrace();

}

**return** employees;

}

Chạy phương thức trong main **t.addEmployee**(**new** File("danhsach.txt"));

Nhập nhân viên:

Id: 9

First name : Huy

Last name : Nguyễn

Age : 25

NET : 3

Position : 4

Apply year : 2019

Depart id : 89

Đã Thoát

DANH SACH EMPLOYEE

STT ID FirstName LastName Age Address Net Position ApplyYear DepartId

-----------------------------------------------------------------

1 0 An Chu 70 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

2 0 Anh Nguyễn 11 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

3 0 Anh Vũ 1 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

4 0 Cường Võ 36 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

5 1 Phương Minh 25 NoStreetName NoDistrictName NoCityName 1 null 3 3

6 9 Huy Nguyễn 25 NoStreetName NoDistrictName NoCityName 3 4 2019 89

1. Phương thức sửa nhân viên có mã nhập vào

//phương thức sửa nhân viên cập nhật với file

@Override

**public** ArrayList<Employee> editEmployee(File f)

{

//Khởi tạo danh sách nhân viên

ArrayList<Employee> employees = **null**;

Employee e = *sua*();

**try** {

// Đọc danh sách nhân viên từ file

employees = MyFile.*textInputFile*(f.getPath());

**for**(**int** i = 0;i<employees.size();i++) {

// Kiểm tra xem mã nhân viên nhập vào có trong file hay không

**if**(employees.get(i).getId()==e.getId()) {

employees.set(i, e);

**break**;

}

}

//Cập nhật lại file

MyFile.*textOutputFile*(f.getPath(), employees);

} **catch** (IOException ex) {

ex.printStackTrace();

}

**return** employees;

}

Chạy phương thức trong main **t.editEmployee**(**new** File("danhsach.txt"));

Nhập nhân viên:

Nhập mã nhân viên cần sửa : 9

Nhập thông tin nhân viên cần sửa : First name : Hoa

Last name : Vũ

Age : 20

NET : 2

Position : 2

Apply year : 2

Depart id : 2

Đã Thoát

DANH SACH EMPLOYEE

STT ID FirstName LastName Age Address Net Position ApplyYear DepartId

-----------------------------------------------------------------

1 0 An Chu 70 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

2 0 Anh Nguyễn 11 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

3 0 Anh Vũ 1 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

4 0 Cường Võ 36 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

5 1 Phương Minh 25 NoStreetName NoDistrictName NoCityName 1 null 3 3

6 9 Hoa Vũ 20 NoStreetName NoDistrictName NoCityName 2 2 2 2

1. Phưởng thức xóa nhân viên theo mã nhập vào

//phương thức xóa nhân viên cập nhật lại file

@Override

**public** ArrayList<Employee> delEmployee(File f)

{

// Nhập mã nhân viên muốn xóa từ bàm phím

Scanner s = **new** Scanner(System.***in***);

**int** id=-1;

**do** {

System.***out***.print("Nhập mã nhân viên muốn xóa: ");

**try** {

id = s.nextInt();

}**catch** (NumberFormatException e) {

System.***out***.println("Mã nhân viên không hợp lệ!");

}

// Nếu mã nhân viên âm thì yêu cầu nhập lại

**if**(id<0)

{

System.***out***.println("Nhập lại mã nhân viên.");

}

}**while**(id<0);

//Khởi tạo danh sách nhân viên

ArrayList<Employee> employees = **null**;

**try** {

/ Đọc danh sách nhân viên từ file

employees = MyFile.*textInputFile*(f.getPath());

**for**(Employee employee:employees)

{

// Kiểm tra xem mã nhân viên nhập vào có trong file hay không

**if**(employee.getId()==id) {

// xóa nhân viên khỏi danh sách và thoát khỏi vòng lặp

employees.remove(employee);

**break**;

}

}

//Cập nhật lại file

MyFile.*textOutputFile*(f.getPath(), employees);

**return** employees;

} **catch** (IOException e) {

e.printStackTrace();

**return** **null**;

}

}

Chạy phương thức trong main t.delEmployee(**new** File("danhsach.txt")), ta thấy nhân viên có **id = 9** đã bị xóa khỏi file

Nhập mã nhân viên muốn xóa: 9

DANH SACH EMPLOYEE

STT ID FirstName LastName Age Address Net Position ApplyYear DepartId

-----------------------------------------------------------------

1 0 An Chu 70 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

2 0 Anh Nguyễn 11 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

3 0 Anh Vũ 1 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

4 0 Cường Võ 36 NoStreetName NoDistrictName NoCityName 0 NONE 2021 0

5 1 Phương Minh 25 NoStreetName NoDistrictName NoCityName 1 null 3 3

HẾT