# Module 1

# Ứng Dụng Quản Lý Khu nghỉ dưỡng Furama

# Mục đích

Học viên sử dụng những kiến thức đã học ở module 1 để viết ứng dụng quản lý Customer, Employees và các dịch vụ của khu nghỉ dưỡng Furama.

# Công nghệ phải dùng

Sau khi kết thúc khóa bài tập , học viên sẽ nắm được các kiến thức sau :

**Javascript**

|  |
| --- |
| Biến và Kiểu dữ liệu (variables) |
| Toán tử và Biểu thức (Operator, expression) |
| Cấu trúc điều kiện (condition structure) |
| Cấu trúc lặp (Loop) |
| Mảng (Array) |
| Hàm/Thủ tục (function/ method) |
| Lập trình hướng đối tượng (OOP) |
| Lớp (class), đối tượng (object), phương thức (method), thuộc tính (attributes), hàm tạo (function) |

# Mô tả hệ thống

Xây dựng hệ thống quản lý khu nghỉ dưỡng Furama tại thành phố Đà Nẵng. Hệ thống được mô tả như sau:

1. Khu nghỉ dưỡng Furama sẽ cung cấp các dịch vụ cho thuê Villa, House, Room, tất cả các dịch vụ này sẽ bao gồm các thông tin: Tên dịch vụ, diện tích sử dụng, chi phí thuê, Số lượng người tối đa, Kiểu thuê (bao gồm thuê theo năm, tháng, ngày, giờ).

* Riêng Villa sẽ có thêm thông tin: Tiêu chuẩn phòng, Mô tả tiện nghi khác, Diện tích hồ bơi, Số tầng.
* Riêng House sẽ có thêm thông tin: Tiêu chuẩn phòng, Mô tả tiện nghi khác, Số tầng.
* Riêng Phòng sẽ có thêm thông tin: Dịch vụ miễn phí đi kèm.

Ngoài ra, resort còn cung cấp các dịch vụ đi kèm như massage, karaoke, thức ăn, nước uống, thuê xe di chuyển tham quan resort.

Dịch vụ đi kèm sẽ bao gồm các thông tin: Tên dịch vụ đi kèm, Đơn vị, Giá tiền

1. Hệ thống quản lý còn cho phép quản lý thông tin của Employee resort. Thông tin Employee sẽ bao gồm: Họ tên Employee, Ngày sinh, Số CMND, Số ĐT, Email, Trình độ, Vị trí, lương

Trình độ sẽ lưu trữ các thông tin: Trung cấp, Cao đẳng, Đại học và sau đại học

Vị trí sẽ lưu trữ các thông tin: Lễ tân, phục vụ, chuyên viên, giám sát, quản lý, giá đốc.

Mỗi Employee sẽ thuộc một bộ phận làm việc: Hiện tại resort bao gồm các bộ phận: Sale – Marketing, Hành Chính, Phục vụ, Quản lý.

1. Hệ thống cho phép quản lý Customer đến thuê và sử dụng các dịch vụ của Furama. Thông tin của Customer bao gồm: Họ tên Customer, Ngày sinh, Giới tính, Số CMND, Số ĐT, Email, Loại khách, Địa chỉ (resort sẽ phân loại Customer để áp dụng các chính sách phụ vụ cho phù hợp).

Loại Customer bao gồm: (Diamond, Platinium, Gold, Silver, Member).

Customer có thể sử dụng các dịch vụ thuê Villa, House, Phòng và các dịch vụ đi kèm.

Đối với các dịch vụ thuê Villa và House thì Customer cần phải làm hợp đồng thuê với Furama

Hợp đồng thuê sẽ bao gồm các thông tin: Số hợp đồng, Ngày bắt đầu, Ngày kết thúc, Số tiền cọc trước, Tổng số tiền thanh toán.

# Yêu cầu hệ thống

## Yêu cầu 1 *(#1. Biến kiểu dữ liệu và toán tử*)

**Objectives**

* *Tạo được các biến và kiểu dữ liệu thông dụng*
* *Biết cách sử dụng alert và prompt*
* *Sử dụng toán tử +, - , \*, /.*
* *Sử dụng hàm ép kiểu.*

Tạo project có tên là Module1\_AccountName để thực hiện các yêu cầu sau:

**Task 1:**

Cho phép người dùng nhập vào tên, số CMND, ngày tháng năm sinh, email, địa chỉ, loại Customer (Diamond, Platinum, Gold, Silver, Member), giảm giá, số lượng đi kèm, số ngày thuê, loại dịch vụ (Villa, House, Room), loại phòng thuê (Vip, Business, Normal).

(***cố gắng đặt tên biến bằng tiếng anh và có ý nghĩa ví dụ: số ngày thuê = rentDays***)

Bảng giá:

* Villa 500$/ 1 ngày
* House 300$/1 ngày
* Room: 100$/1 ngày

**Task 2:**

Hiển thị toàn bộ thông tin vừa nhập kèm theo giá sau khi đã được giảm giá

( Số tiền phải trả = giá loại dịch vụ X số ngày X ( 1 – giảm giá/100 ) )

(*gán giá trị trực tiếp cho giá loại dịch vụ từ bảng giá bên trên*)

***Có thể xem video demo để hiểu cách thực hiện***

## Yêu cầu 2 *(#2. Cấu trúc điều kiện)*

**Objectives**

* *Sử dụng được câu lệnh if, if else, else if , if lồng nhau*
* *Sử dụng được swich case*
* *Biết cách xử lý ngày tháng bằng 1 số hàm trong JS*

Để thuận tiện thao tác với các chức năng của hệ thống cần xây dựng 1 trình đơn (menu) cho phép lựa chọn các chức năng.

Hãy tạo sẵn thông tin cho 1 Customer như yêu cầu 1 bằng cách thực hiện thêm dữ liệu cho Customer đó trực tiếp trên mã nguồn.

Nếu người dùng chọn sai (*các lựa chọn không nằm trong menu*) hiển thị “Failed”

*(Lưu ý: Copy bài trước ra 1 bài mới để có thể xem lại code từng ngày dễ dàng hơn)*

1. **Show Information Customer.**

Hiển thị toàn bộ thông tin của Customer.

1. **Edit Information Customer.**

Cho phép chỉnh sửa thông tin của Customer. **Sau đó hiển thị lại toàn bộ thông tin của Customer sau khi đã được chỉnh sửa**

(*Sau khi chọn edit hiển thị 1 menu con show thuộc tính và giá trị của thuộc tính cho phép người dùng lựa chọn thuộc tính muốn chỉnh sửa, nếu sự lựa chọn của người dùng không nằm trong menu con hiển thị “Failed”*)

1. **Show the amount to be paid after discount**

*( Số tiền phải trả = giá loại dịch vụ X số ngày X ( 1 – giảm giá/100 )*

*ở bài này giá loại dịch vụ sẽ phụ thuộc vào loại dịch vụ, giả sử gán loại dịch vụ là Villa chương trình sẽ thực hiện gán giá loại dịch vụ là 500 tương tự house 300 và room 100, hãy viết câu lệnh if để thực hiện yêu cầu này )*

Hiển thị số tiền Customer phải chi trả (được tính theo công thức bên trên). Số tiền Customer cần chi trả sẽ được tính dựa trên các điều kiện sau *(sau khi thực hiện công thức bên trên hãy xét thêm điều kiện bên dưới để đưa ra số tiền phải trả của Customer)*:

Mức giá theo địa chỉ:

* Đà Nẵng giảm giá 20$
* Huế giảm giá 10$
* Quảng Nam giảm giá 5$

Mức giá theo thời gian lưu trú:

* 7 ngày trở lên giảm giá 30$
* Từ 5 – 7 ngày giảm giá 20$
* Từ 2 – 4 ngày giảm giá 10$

Mức giá theo loại Customer:

* Nếu loại Customer là Diamond giảm giá 15$
* Nếu loại Customer là Platium giảm giá 10$
* Nếu loại Customer là Gold giảm giá 5$
* Nếu loại Customer là Silver giảm giá 2$
* Nếu loại Customer là Member thì không giảm giá

Mức giá dựa vào tuổi và địa chỉ:

* Tuổi từ 30 trở lên và địa chỉ tại Đà Nẵng thì giảm giá 2$
* Tuổi từ 20 đến 30 tuổi và địa chỉ tại Đà Nẵng thì giảm giá 1$

**Hướng dẫn:**

*Để xử lý ngày tháng để tính tuổi tham khảo mã sau:*

let currentYear = new Date().getFullYear();

                let birthday = new Date(birthdayCustomer).getFullYear();

                let ageCustomer = current - birthday;

***Có thể xem video demo để hiểu cách thực hiện***

## Yêu cầu 3 *(#3. Cấu trúc lặp)*

**Objectives**

* *Sử dụng câu lệnh lặp do while, while, for*
* *Sử dụng Break, Continue*
* *Sử dụng hàm isNaN, Number.isInteger trong JS*
* *Sử dụng 1 số hàm xử lý chuỗi subString, toLowerCase, toUpperCase, trim.*

*(Lưu ý: Copy bài trước ra 1 bài mới để có thể xem lại code từng ngày dễ dàng hơn)*

*(Ở bài này chúng ta sử dụng copy lại mã nguồn ở yêu cầu 1 để thực hiện task sau đây)*

**Task 1:**

Để kiểm tra tính hợp lệ dữ liệu do người dùng nhập vào thì hệ thống cần xét các trường hợp sau: Áp dụng lệnh while hoặc do-while để yêu cầu người dùng phải nhập lại khi nhập dữ liệu sai định dạng.

* (\*) Địa chỉ email phải đúng theo định dạng [abc@abc.abc](mailto:abc@abc.abc)
* Số CMND phải đúng theo định dạng XXXXXXXX với X là số nguyên dương
* (\*) Ngày sinh phải đúng theo định dạng DD/MM/YYYY

(*Sử dụng boolean, for và do while để duyệt và kiểm tra tính hợp lệ cho trường hợp này*)

* Giảm giá, số ngày thuê và người đi kèm phải là số nguyên dương

*Ví dụ*: Khi nhập giá trị cho thuộc tính có kiểu dữ liệu là số nguyên thì người dùng phải nhập dữ liệu phải là kiểu số thì hệ thống mới cho phép nhập tiếp các thuộc tính khác hoặc tiến hành các bước tiếp theo.

**Task 2:**

Áp dụng vòng lặp for để xử lý xóa bỏ tất cả khoảng trắng thừa trong tên của Customer, địa chỉ, loại khách hàng và chuẩn hóa dữ liệu.

(*Giả sử nhập tên ngUYEn thANH KIEN hãy xử lý sao cho kết quả hiển thị thông tin sau khi nhập ở mục Name Customer là Nguyen Thanh Kien*)

***Có thể xem video demo để hiểu cách thực hiện***

## Yêu cầu 4 *(#4. Mảng)*

**Objectives**

* *Sử dụng được mảng 1 chiều thành thạo*
* *Sử dụng được vòng lặp để duyệt mảng*
* *Sử dụng được các thao tác cơ bản với mảng*
* *Biết thêm một số kĩ thuật xử lý mảng*

***(Lưu ý: Ở bài này không cần copy lại mã nguồn cũ, tạo ra file YeuCau4.html và tiến hành thực hiện các task sau)***

Hệ thống cần lưu được thông tin của nhiều Customer. Áp dụng mảng để thực hiện lưu thông tin nhiều Customer.

**Task 1:**

1. Tạo menu gồm có tính năng Add New Customer và Delete Customer

(*sau khi người dùng chọn Add New Customer hãy hiển thị ô nhập số lượng customer muốn thêm vào sau đó tiến hành nhập tên customer và hiển thị ra toàn bộ tên của customer có trong mảng arrNameCustomers*)

1. Xuất ra mảng arrNameCustomers được sắp xếp theo thứ tự alphabel

(*Sau khi hiển thị ra toàn bộ tên của Customer có trong mảng ở bên trên tiếp ngay sau đó hãy hiển thị ra yêu cầu này*)

**Task 2:**

Hãy thêm trực tiếp vào mã nguồn để có 1 mảng gồm 10 phần tử là tên của Customer. Sau đó từ Menu hãy chọn Delete Customer vào hiển thị ra menu con chứa danh sách tên của các Customer, chọn 1 Customer muốn xóa và in ra tên của toàn bộ các Customer sau khi xóa.

***Có thể xem video demo để hiểu cách thực hiện***

## Yêu cầu 5 *(#5. Hàm)*

**Objectives**

* *Khai báo và sử dụng cơ bản về hàm*
* *Khai báo và sử dụng được hàm có tham số*
* *Hiểu biết và sử dụng 1 số kĩ thuật về hàm*
* *Xây dựng tư duy lập trình và tư duy logic về luồng dữ liệu*
* *(\*) Hiểu được tầm quan trọng về hiệu ứng domino giữa các tính năng (Important)*

*(Lưu ý: Ở bài này chúng ta không copy lại mã nguồn cũ mà sẽ tổng hợp kiến thức và mã nguồn của các yêu cầu trước để hoàn thành các task bên dưới)*

**Task 1:** Thực hiện các yêu cầu sau:

1. Xây dựng function tên displayMainMenu() hiển thị menu để người dùng lựa chọn các tính năng

* Add New Customer.
* Display Information Customer.
* Edit Information Customer.
* Delete Customer.
* Exit.

1. Xây dựng hàm addNewCustomer() để thực hiện thêm thông tin của 1 khách hàng với các thuộc tính như yêu cầu 1 vào một mảng 2 chiều có tên là arrListCustomers. (*Sau khi thực hiện xong trả về Main Menu*)
2. Xây dựng hàm displayCustomers() (*Sau khi chọn vào đây hãy hiển thị 1 menu con chứa danh sách toàn bộ tên và số CMND của các Customer* ***và 1 lựa chọn back.*** *Nếu người dùng chọn Customer nào thì hiển thị toàn bộ thông tin của Customer đó thông qua hàm được gợi ý bên dưới, nếu chọn back thì quay về menu trước đó*). Sau khi thực hiện xong trả về Main Menu.

**Note:** Ở bài này hãy xây dựng 1 hàm displayInfomationCustomer() để hiển thị toàn bộ thông tin của 1 khách hàng bất kì sau đó sử dụng hàm này vào bên trong của hàm displayCustomers().

1. Xây dựng fuction tên displayInfomationCustomer() để hiển thị toàn bộ thông tin của Customer

**Task 2:**

1. Định nghĩa function có tên là editCustomer để chỉnh sửa thông tin cho Customer, sau khi chỉnh sửa hiển thị lại thông tin của Customer đó sau đó trả về **main menu**

(*sau khi chọn edit hiển thị 1 menu con show toàn bộ tên Customer và IdCard cho phép người dùng chọn 1 Customer tiếp theo hiển thị 1 menu show thuộc tính và giá trị của thuộc tính cho phép người dùng lựa chọn thuộc tính muốn chỉnh sửa*)

**Note:** Trong các menu con luôn có 1 lựa chọn back để quay về menu gần nhất trước đó. Tận dụng lại hàm displayCustomers() ở bên trên.

1. Tương tự cho phần Delete Customer (*làm như phần note của editCustomer*)

**Note:** Sau khi chọn Customer muốn xóa hãy hiển thị thông báo yêu cầu xác nhận 1. Yes 2. No nếu chọn 1 sẽ tiến hàng xóa và trả về Main Menu, nếu chọn 2 sẽ không tiến hành xóa và trả về Main Menu

***Có thể xem video demo để hiểu cách thực hiện***

## Yêu cầu 6

**Objectives**

* *Khai báo và sử dụng được lớp*
* *Khai báo và sử dụng được đối tượng*
* *Xây dựng và sử dụng được phương thức và thuộc tính của đối tượng*
* *Biết cách sử dụng Regular Expression*

*(Lưu ý: Copy bài trước ra 1 bài mới để có thể xem lại code từng ngày dễ dàng hơn)*

**Task 1:**

Thực hiện các yêu cầu sau:

1. Tạo đối tượng có tên là Customer bao gồm các thuộc tính ở phần mô tả như yêu cầu 1
2. Tiếp tục khởi tạo các phương thức getter/setter cho các thuộc tính của Customer
3. Định nghĩa thêm phương thức tính tiền phòng cho các Customer
4. Định nghĩa lại hàm addNewCustomer để thêm mới Customer vào mảng có tên là arrCustomers.

Chức năng thêm mới Customer sẽ cho phép người sử dụng tạo mới thông tin cho 1 Customer từ trình đơn (menu) đã định nghĩa từ Yêu cầu 1.

**Task 2:**

Thực hiện các yêu cầu sau:

1. Tạo đối tượng có tên là Employee bao gồm các thuộc tính ở phần mô tả như yêu cầu 1
2. Tiếp tục khởi tạo các phương thức getter/setter cho các thuộc tính của Employee
3. Định nghĩa thêm phương thức tính tiền phòng cho các Employee
4. Định nghĩa lại hàm addNewEmployee để thêm mới Employee vào mảng có tên là arr Employees.

Chức năng thêm mới Customer sẽ cho phép người sử dụng tạo mới thông tin cho 1 Customer từ trình đơn (menu) đã định nghĩa từ Yêu cầu 1.

Xây dựng phương thức tính lương cho từng Employee theo điều kiện sau đây

* Trợ cấp 500$ cho Employee nào làm ở vị trí Manager
* Trợ cấp 300$ cho Employee nào làm ở vị trí Sale
* Trợ cấp 200$ cho Employee làm việc ở vị trí Marketing

**Task 3:**

Định nghĩa lại function addNewCustomer() có kiểm trả tính hợp lệ của dữ liệu trước khi thực hiện thêm mới theo những yêu cầu sau đây:

* Email phải có định dạng [abc@abc.com](mailto:abc@abc.com)
* Ngày tháng năm sinh phải đúng chuẩn dd/MM/yyyy
* Những thuộc tính là số thì không cho phép nhập vào kí tự
* Số chứng minh nhân dân phải có đúng số và không được nhập kí tự

***Có thể xem video demo để hiểu cách thực hiện***

# Module 2 (sử dụng bài toán mô tả ở module1)

# Yêu cầu 7

Objectives

* *Khai báo và sử dụng được 4 tính chất* Encapsulation, Abstraction, Inheritance, Polymorphism
* *Access modifier*
* *Constructor*
* *Static, final*
* *Overload, Override*
* *Sử dụng abstract class*
* *Sử dụng được interface class*
* *Sử dụng được quan hệ is-a, has-a*
* *Sử dụng được các từ khóa super, this*
* *Quản lý được các Exception*
* *Sử dụng được các lớp Java Utilities*
* *Sử dụng được Java collection*
* *Sử dụng jdbc driver để kết nối cơ sở dữ liệu*

**Task 1**: Học viên thiết kế các class của chương trình tuân thủ theo đúng mô hình OOP theo hướng dẫn áp dụng đầy đủ 4 tính chất: bao đóng (*encapsulation)* , kế thừa (*inheritance)* , đa hình (*polymorphism)* , trừu tượng (*abstraction).* Để thực hiện yêu cầu sau:

* + - 1. Xây dựng abstract class Services (dịch vụ) bao gồm các thông tin chung của tất cả dịch vụ cho thuê Villa, House, Room.
      2. Xây dựng abstract method có tên là **showInfor()** trong Services class để hiển thị thông tin của mỗi dịch vụ cho thuê.
      3. Xây dựng các class có tên là Villa, House, Room và định nghĩa thêm các thuộc tính riêng của từng class. Các class này sẽ extend từ abstract class có tên là Services.

*Lưu ý: sử dụng super keyword để sử dụng các thuộc tính đã được định nghĩa trong Services abstract class*.

* + - 1. Thực hiện override phương thức **showInfor()** tương ứng với mỗi class Villa, House, Room để hiển thị thông tin phù hợp với từng dịch vụ thê nhà.

**Task 2:** Học viên viết hàm có tên là addNewServies để thực hiện nhập thông tin cho một dịch vụ cho thuê bất kỳ (Villa, House hoặc Room). Các thông tin này sẽ được lưu vào database

*Lưu ý: database tham khảo tại video.*

**Task 3:** Sau khi nhập xong thông tin các dịch vụ thuê nhà vào database thì hiển thị số lượng các dịch vụ cho thuê ra màn hình.

*Lưu ý: có thể tạo một biến static có tên services\_count ở Services class để thực hiện đếm số lượng dịch vụ cho thuê đối với 1 lần nhập của người dùng.*

**Task 4:**

Viết hàm kiểm tra dữ liệu đầu vào khi người dùng thực hiện thêm mới một dịch vụ cho thuê. Yêu cầu kiểm tra được hợp lệ của các thuộc tính:

Diện tích sử dụng phải là số nguyên và lớn hơn 30m2

Chi phí thuê phải là số dương

Số lượng người tối đa phải >0 và nhỏ hơn <20

Dịch vụ đi kèm phải là các giá trị: massage, karaoke, food, drink, car

Số tầng phải số nguyên dương.

**Task 5:**

* + - 1. Tạo class có tên là Customer với các thuộc tính như sau: Họ tên Customer, Ngày sinh, Giới tính, Số CMND, Số ĐT, Email, Loại khách, Địa chỉ và thuộc tính sử dụng dịch vụ có kiểu đối tượng là Services.
      2. Sử dụng cơ sở dữ liệu gợi ý trong video để thực hiện viết hàm có tên là addNewCustomerForRen() cho phép chương trình thêm mới một khách hàng thuê dịch vụ nhà bất kỳ (có thể là Villa, House hoặc Room) lưu vào database.
      3. Tạo user Exception cho các trường hợp
* EmailException để throw ra màn hình trong trường hợp dữ liệu nhập vào không đúng định dạng email
* DateException để throw ra màn hình trong trường hợp dữ liệu nhập vào không đúng định dạng ngày sinh. (bắt luôn cho trường hợp nhập năm sinh < 1900).
* IdException để throw ra màn hình trong trường hợp dữ liệu nhập vào không đúng định sạng chứng minh nhân dân (phải là 9 số nguyên dương)

Ngoài ra, nếu có bất kỳ một exception nào khác trong quá trình nhập thông tin khách hàng thuê nhà thì phải catch được và thông báo với nội dung như sau: “*The system has encountered an unexpected problem*!!!”

Nếu thõa mãn các yêu cầu về catch exception thì cho phép lưu vào database.

**Task 6:**

Sử dụng showInfor() đã được thiết kế để hiển thị toàn bộ dịch vụ cho thuê ra màn hình theo định dạng sau:

….. thiết kế sau…..

Lưu ý: Sử dụng ArrayList để thực hiện yêu cầu này

Khi hiển thị danh sách các dịch vụ cho thuê ra màn hình thì phải sắp xếp theo tăng dần theo thuộc tính Tên dịch vụ, nếu tên dịch vụ trùng nhau thì nếu dịch vụ nào có Chi phí lớn hơn sẽ đứng trước.

Lưu ý: học viên sử dụng hàm sort có trong collection để thực hiện yêu cầu này.

**Task 7**: Giả sử khi truy vấn các dịch vụ cho thuê ra màn hình nhưng tên dịch vụ bị trùng lặp thì thực hiện loại bỏ các bản ghi trùng lặp bằng cách sử dụng List, Set hoặc Map

Sau đó hiển thị ra màn hình.

Task 8: (Sử dụng Map)

Furama hiện tại đã có 10 nhân viên làm việc . Mỗi nhân viên đều có 1 mã số duy nhất ứng với tên của mình. Ví du Mã số 001 Tên Nhân Viên là Nguyễn Thị B.

Thực hiện yêu cầu sau:

1. Tạo ra 10 nhân viên có tên tuổi địa chỉ
2. Add 10 nhân viên vào mỗi cái Map
3. Cái key sẽ là mã số nhân viên còn value sẽ là Object nhân viên
4. Viết 1 hàm in ra toàn bộ nhân viên có trong Furama.

Task 9: (Sử dụng HashSet)

Hằng ngày có rất nhiều chuyến xe ra vào Furama. Chúng ta cần một nơi để lưu trữ số lần có mỗi chiếc xe du lich vào Furama. Ví dụ có 1 chiếc xe vào thì mình lưu lại bản số xem và cuối ngày mình xem có bao nhiêu xe đi vào Furama

Thực hiện yêu cầu sau:

1. Tạo Set collection để lưu trữ
2. Khi có một xe vào thì add vào Set
3. Viết 1 hàm để duyệt qua các biển số xe trong ngày

Task 10: (Sử dụng Queue)

Furama có mở thêm rạp chiếu phim 4D cho khách hàng. Khi khách hàng mua vé trước thì được xếp vào hàng ghế đầu tiên, nếu khách hàng mua cuối cùng thì ngồi hàng ghế sau cùng.

Thực hiện yêu cầu sau:

1. Tạo một Queuu tập hợp
2. Khi có 1 khách hàng vào đầu tiên , thì dùng Queue add vào hàng đợi
3. Sau khi bán hết vé xong thì in ra danh sách các khách hàng mua vé hôm đó theo thứ tự người mua trước sẽ được in trước
4. Viết method để in ra danh sách

Task 11: (Sử dụng Stack)

Bộ phận nhân sự của Furama cần 1 cái tủ để đựng hồ sơ của nhân viên. Khi có nhân viên nào muốn tìm hồ sơ của mình thì bộ phận nhân sự sẽ lục trong tủ hồ sơ

Thực hiện yêu cầu sau:

1. Tạo một class là Tủ Hồ Sơ
2. Sau đó dùng Stack để lưu trử các hồ sơ
3. Viết một hàm tìm kiếm hồ sơ nhân viên

Task 12: (Sử dụng File)

Hiện nay công việc kinh doanh khá ổn nhưng kế toán cần phải có tài liệu để kiểm tra tháng này có bao nhiêu khách hàng đặt phòng trong khu nghỉ dưỡng. Do vậy mình phải lưu trữ thông tin của Khách hàng vào một file.

1. Tạo một method lưu tất cả các Khách hàng xuống 1 file