

## **Module 2**

# **Ứng Dụng Quản Lý Khu nghỉ dưỡng Furama**

## **I. 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.

## **II. Công nghệ phải dùng**

### III. 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 bao gồm Villa, House, Room.

Tất cả các dịch vụ này sẽ bao có 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

2. Hệ thống quản lý còn cho phép quản lý thêm 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.

3. Hệ thống cho phép quản lý thêm thông tin của các 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 Điện Thoại, Email, Loại khách, Địa chỉ (resort sẽ phân loại Customer để áp dụng các chính sách phục vụ cho phù hợp).

Loại Customer bao gồm: (Diamond, Platinum, 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.

# IV. Yêu cầu hệ thống

## Yêu cầu

### 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

Để thực hiện các task dưới đây trước hết hãy tạo Project có tên FuramaResort, sau đó trong mục src hãy tạo các package sau:

Models, Views, Controllers, Commons,Data,Libs

**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.

(trong class Services lưu ý thêm thuộc tính id kiểu dữ liệu String và các class này được tạo trong package models)

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.

(Các class này được tạo trong package models)

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

4. 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 thuê nhà.
5. Sau đó sử dụng phương thức **showInfor()** để hiển thị thông tin của đối tượng vừa được tạo.

## **Task 2:**

Tạo 1 class có tên **MainController** nằm trong package **Controllers** sau đó thực hiện yêu cầu sau:

1. Học viên tạo phương thức **displayMainMenu()** để hiển thị các trình menu cho phép người dùng lựa chọn các chức năng trên menu. Gồm có:
  1. Add New Services
  2. Show Services
  3. Add New Customer
  4. Show Information of Customer
  5. Add New Booking
  6. Show Information of Employee
  7. Exit
  - Khi chọn chức năng số 1 Add New Services thì chương trình sẽ gọi phương thức **addNewServices()** được mô tả ở mục 2
  - Khi chọn chức năng số 2 Show Services thì chương trình sẽ gọi phương thức **showServices()** ở Task 3.
  - Khi chọn chức năng số 3 Add New Customer thì chương trình sẽ gọi phương thức được mô tả ở Task 5
  - Khi chọn chức năng số 4 Show Information of Customer thì chương trình sẽ gọi phương thức được mô tả ở Task 5
  - Khi chọn chức năng số 5 Add New Booking thì chương trình sẽ gọi phương thức được mô tả ở Task 7
  - Khi chọn chức năng số 6 Show Information of Employee thì chương trình sẽ gọi phương thức được mô tả ở Task 9
2. Học viên tạo phương thức có tên là **addNewServices()** để thực hiện nhập thông tin cho một dịch vụ cho thuê bất kỳ (Villa, House hoặc Room).

Bao gồm trình đơn:

1. Add New Villa
  2. Add New House
  3. Add New Room
  4. Back to menu
  5. Exit
- Khi người dùng chọn chức năng số 1 thì chương trình sẽ cho phép nhập các thuộc tính của Villa và sau đó ghi vào file data/Villa.csv
  - Khi người dùng chọn chức năng số 2 thì chương trình sẽ cho phép nhập các thuộc tính của House và sau đó ghi vào file data/House.csv
  - Khi người dùng chọn chức năng số 3 thì chương trình sẽ cho phép nhập các thuộc tính của Room và sau đó ghi vào file data/Room.csv

Ghi chú: Học viên tạo class có để thực hiện đọc/ghi file csv

### **Task 3:**

- Học viên xây dựng phương thức showServices() hiển thị trình đơn bao gồm:
  1. Show all Villa
  2. Show all House
  3. Show all Room
  4. Show All Name Villa Not Duplicate
  5. Show All Name House Not Duplicate
  6. Show All Name Name Not Duplicate
  7. Back to menu
  8. Exit
- Chọn chức năng 1 Show all Villa thì chương trình sẽ hiển thị danh sách tất cả các Villa
- Chọn chức năng 2 Show all House thì chương trình sẽ hiển thị danh sách tất cả các House
- Chọn chức năng 3 Show all Room thì chương trình sẽ hiển thị danh sách tất cả các Room
- Chọn chức năng 4 Show all name Villa not duplicate thì chương trình sẽ hiển thị danh sách tên tất cả các Villa có trong hệ thống được mô tả ở Task 8
- Chọn chức năng 5 Show all name House not duplicate thì chương trình sẽ hiển thị danh sách tên tất cả các House có trong hệ thống được mô tả ở Task 8
- Chọn chức năng 6 Show all name Room not duplicate thì chương trình sẽ hiển thị danh sách tên tất cả các Room có trong hệ thống được mô tả ở Task 8
- Chọn chức năng 7 Back to menu thì chương trình sẽ hiển thị trình đơn được xử lý ở **displayMainMenu()**
- Chọn chức năng 8, thoát khỏi chương trình

*Lưu ý: Sử dụng **ArrayList** để lấy dữ liệu tương ứng từ File CSV tương ứng của từng mục, sau đó sử dụng **foreach** để duyệt **arraylist** kèm theo phương thức **showInfor()** của mỗi đối tượng đã được tạo ở bài trước để show thông tin ra màn hình.*

### **Task 4:**

Viết hàm kiểm tra dữ liệu đầu vào khi người dùng thực hiện thêm mới một dịch vụ (add new Services) cho thuê. Yêu cầu kiểm tra được hợp lệ của các thuộc tính: (**sử dụng regular expression**)

- Mã dịch vụ phải đúng định dạng: SVXX-YYYY, với YYYY là các số từ 0-9, XX là:
  - Nếu là Villa thì XX sẽ là VL
  - Nếu là House thì XX sẽ là HO
  - Nếu Room thì XX sẽ là RO
- Tên dịch vụ phải viết hoa ký tự đầu, các ký tự sau là ký tự bình thường
- Diện tích sử dụng và diện tích hồ bơi phải là số thực lớn hơn 30m<sup>2</sup>
- 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 là số nguyên dương.
- Kiểu thuê, Tiêu chuẩn phòng chuẩn hóa dữ liệu giống tên dịch vụ
- Ngày sinh phải nhỏ hơn ngày hiện tại 18 năm và phải đúng định dạng dd/mm/YYYY

*Trong trường hợp bị lỗi phải thông báo lỗi và bắt người dùng nhập lại cho tới khi thỏa mãn yêu cầu.*

### **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, và phương thức **showInfor()**.
2. Xây dựng phương thức **addNewCustomer()** cho phép người dùng nhập thông tin cho Customer sau đó lưu vào file Customer.CSV.
3. Xây dựng phương thức **showInformationCustomers()** cho phép người dùng sau khi chọn sẽ hiển thị ra toàn bộ thông tin của các Customer có trong file Customer.CSV.
4. Tạo user Exception để valid các trường hợp sau:
  - **Tạo NameException để kiểm tra Name Customer** phải có định dạng in hoa ký tự đầu tiên và không cho phép ký tự khoảng trống thừa. Ví dụ “Nguyễn Kiên” là đúng còn “Nguyễn KIÊN” là sai. Nếu không hợp lệ thì sẽ hiển thị thông báo “Tên Khách hàng phải in hoa ký tự đầu tiên trong mỗi từ”.
  - **Tạo EmailException để kiểm tra, Email** phải nhập đúng kiểu dữ liệu của email ở mức độ tương đối. **Email** bao gồm duy nhất chữ @ và tối thiểu 1 dấu . , có thể suy nghĩ và phát triển thêm. Nếu không hợp lệ thì sẽ hiển thị thông báo “Email phải đúng định dạng abc@abc.abc”.
  - **Tạo GenderException để** bắt buộc người dùng phải nhập vào Male, Female hoặc Unknow. Trong trường hợp người dùng nhập sai in hoa hoặc in thường vẫn chấp nhận nhưng phải chuẩn hóa dữ liệu đúng trước khi lưu vào file.  
Ví dụ: uNKNOW, maLE vẫn chấp nhận và chuẩn hóa dữ liệu thành Unknow và Male sau đó lưu vào File.
  - **Tạo IdCardException kiểm tra Id Card** phải gồm có 9 chữ số theo định dạng XXX XXX XXX. Nếu sai thì hiển thị thông báo “Id Card phải có 9 chữ số và theo định dạng XXX XXX XXX”.
  - **Tạo BirthdayException để kiểm tra Birthday** phải có định dạng đúng ngày sinh theo kiểu dd/MM/yyyy và có năm sinh >1900 và nhỏ hơn ngày hiện tại là 18 năm. Nếu sai thì hiển thị thông báo lỗi “Năm sinh phải >1900 và nhỏ hơn năm hiện tại 18 năm, đúng định dạng dd/mm/yyyy”.

*Trong trường hợp bị lỗi phải thông báo lỗi và bắt người dùng nhập lại cho đến khi thỏa mãn yêu cầu. Nếu thỏa mãn các yêu cầu trên mới cho phép lưu vào file tránh lưu dữ liệu sai định dạng.*

### **Task 6:**

Cài đặt lại phương thức **showInformationCustomer()** để dữ liệu xuất ra được sắp xếp theo thứ tự Alpha B theo tên của từng Customer.

#### **Trường hợp Customer nào trùng tên thì sắp xếp tăng dần theo năm sinh**

Lưu ý: học viên sử dụng hàm sort (Icomparer, Comparable) có trong collection để thực hiện yêu cầu này.



Tham khảo tại : <https://viblo.asia/p/java-collections-sap-xep-collections-naQZRgvdIvx>

### **Task 7**

- Học viên tạo chức năng Add New Book để cho phép Khách hàng thực hiện booking (đặt dịch vụ).

Phương thức addNewBooking() sẽ cho phép hiển thị danh sách Customer trong file Customer.CSV được đánh số thứ tự để thực hiện việc lựa chọn booking cho Customer nào và trình đơn sau

1. Booking Villa
  2. Booking House
  3. Booking Room
- Khi người dùng chọn chức năng số 1 Booking Villa chương trình sẽ hiển thị danh sách Villa đang có trong hệ thống
  - Khi người dùng chọn chức năng số 2 Booking House chương trình sẽ hiển thị danh sách House đang có trong hệ thống
  - Khi người dùng chọn chức năng số 3 Booking Room chương trình sẽ hiển thị danh sách Room đang có trong hệ thống

**Sau khi chọn xong lưu và file Booking.CSV (sử dụng model Customer đã có sẵn thuộc tính services để lưu thông tin sau đó đẩy vào File Booking.CSV)**

### **Task 8:**

- Tạo phương thức hiển thị danh sách tất cả các tên của Villa không trùng nhau
- Tạo phương thức hiển thị danh sách tất cả các tên của House không trùng nhau
- Tạo phương thức hiển thị danh sách tất cả các Room không trùng nhau

Để hiển thị danh sách tên các Villa, House, Room không trùng nhau thì sử dụng các collection: Set, HashSet, TreeSet để hiển thị danh sách tên ứng với từng dịch vụ sao cho trong danh sách đó không có tên nào trùng nhau và các tên này được hiển thị theo thứ tự tăng dần theo bảng chữ cái alpha b (Nên sử dụng TreeSet).

Link tham khảo TreeSet:

<https://freetuts.net/treeset-trong-java-1096.html>

Link tham khảo đọc ghi file từng dòng để lấy ra name:

<https://gpcoder.com/3107-doc-ghi-file-csv-trong-java/>

***Sau khi thực hiện xong quay về Main Menu.***

### **Task 9:** (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 dụ Mã số 001 Tên Nhân Viên là Nguyễn Thị B.

Tạo 1 class tên là Employee gồm: Họ tên Employee ,tuổi , địa chỉ và phương thức toString().

Thực hiện yêu cầu sau:

1. Tạo ra 10 nhân viên có tên tuổi địa chỉ đọc từ file data/Employee.csv
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. (chức năng số 6)

Link tham khảo Map: <https://viettuts.vn/java-collection/map-trong-java>

### **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 Queue 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 trữ các hồ sơ
3. Viết một hàm tìm kiếm hồ sơ nhân viên

-----HẾT-----