# Module 2 – APJ

# Ứng dụng quản lý khu nghỉ dưỡng Furama

# Mục đích

Học viên sử dụng những kiến thức đã học ở module 1 để viết phần mềm cho khu nghỉ dưỡng Furama. Ứng dụng bao gồm các chức năng về quản lý Customer, Employee, các dịch vụ và các chức năng khác trong khu nghỉ dưỡng

# Công nghệ phải dùng

* Java basic
* Java OOP
* Exception
* Java Collection Framework
* Java Utilities

# 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 loại dịch vụ cho thuê bao gồm Villa, House, Room.

Các loại dịch vụ này sẽ bao có các thông tin: Mã dịch vụ, 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 loại Villa sẽ có thêm thông tin: Tiêu chuẩn phòng, Diện tích hồ bơi, Số tầng.
* Riêng loại House sẽ có thêm thông tin: Tiêu chuẩn phòng, Số tầng.
* Riêng loại Room sẽ có thêm thông tin: Dịch vụ miễn phí đi kèm.

1. Hệ thống quản lý còn cho phép quản lý thêm thông tin của Employee. Thông tin Employee sẽ bao gồm: Mã nhân viên, Họ tên, Ngày sinh, Giới tính, Số CMND, Số Điện Thoại, 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, giám đốc.

1. 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: Mã khách hàng, Họ tên, 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 phải booking để sử dụng các dịch vụ thuê Villa, House, Room và các dịch vụ đi kèm, thông tin booking bao gồm: mã booking, ngày booking, ngày bắt đầu thuê, ngày kết thúc thuê, mã khách hàng, mã dịch vụ.

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, mã booking, số tiền cọc trước, tổng số tiền thanh toán.

Hằng năm Furama sẽ tổ chức 2 sự kiện thường niên là hội nghị khách hàng để tri ân các khách hàng đã sử dụng dịch vụ Villa và House năm vừa qua và chương trình gửi tặng voucher cho khách hàng booking sớm hằng tháng.

Để bảo trì bảo dưỡng các dịch vụ Furama cũng có quy định mỗi một dịch vụ khi sử dụng liên tục 5 lần sẽ được đội kỹ thuật kiểm tra bảo dưỡng các trang thiết bị

# 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:

model, view, controller, utils, service, data, repository

**Task 1:** (task này chỉ yêu cầu tạo menu chương trình sử dụng các vòng lặp, không yêu cầu viết code các chức năng của chương trình)

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

1. Xây dựng menu chương trình với method **displayMainMenu()** để cho phép người dùng lựa chọn các chức năng trên menu. Gồm có:
   1. Employee Management
   2. Customer Management
   3. Facility Management
   4. Booking Management
   5. Promotion Management
   6. Exit

* Khi chọn chức năng số 1 Employee Management thì chương trình hiển thị thanh menu con bao gồm:
  + - 1. Display list employees
      2. Add new employee
      3. Edit employee
      4. Delete employee
      5. Search by name employee
      6. Return main menu
* Khi chọn chức năng số 2 Customer Management thì chương trình hiển thị thanh menu con bao gồm:
  + - 1. Display list customers
      2. Add new customer
      3. Edit customer
      4. Delete customer
      5. Search by name customer
      6. Return main menu
* Khi chọn chức năng số 3 Facility Management thì chương trình hiển thị thanh menu con bao gồm:
  + - 1. Display list facility
      2. Add new facility
      3. Display list facility maintenance
      4. Delete facility
      5. Return main menu
* Khi chọn chức năng số 4 Booking Managerment thì chương trình hiển thị thanh menu con bao gồm:
  + - 1. Add new booking
      2. Display list booking
      3. Create new contracts
      4. Display list contracts
      5. Edit contracts
      6. Return main menu
* Khi chọn chức năng số 5 Promotion Management thì chương trình hiển thị thanh menu con bao gồm:
  + - 1. Display list customers use service
      2. Display list customers get voucher
      3. Return main menu
* Khi chọn chức năng số 6 chương trình sẽ kết thúc

**Lưu ý:** Xử lý Exception cho phần nhập liệu chọn các chức năng của menu chính, menu con

**Task 2**: (Task này chỉ yêu cầu tạo các class, interface, method, property theo OOP)

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 Facility (cơ sở vật chất) bao gồm các thông tin chung của tất cả dịch vụ cho thuê Villa, House, Room.
      2. 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 Facility.
      3. Tương tự 2 mục trên xây dựng class Customer, Employee kế thừa từ abstract class Person
      4. Xây dựng class Booking, Contract

(Các class này được tạo trong package model)

* + - 1. Xây dựng interface Service với các method abstract chung cho các class
      2. Xây dựng các interface IFacilityService, ICustomerService, IEmployeeService, IBookingService, IContactService kế thừa interface Service và định nghĩa thêm các method riêng nếu có
      3. Xây dựng các class triển khai các interface trong mục 6 ở trên như: FacilityService, CustomerService, EmployeeService... (có thể để trống các class triển khai chưa cần code chức năng)

(Các class này được tạo trong package service)

* + - 1. Xây dựng các interface IFacilityRepository, ICustomerRepository, IEmployeeRepository, IBookingRepository, IContactRepository kế thừa interface Repository và định nghĩa thêm các method riêng nếu có
      2. Xây dựng các class triển khai các interface trong mục 6 ở trên như: FacilityRepository, CustomerRepository, EmployeeRepository... (có thể để trống các class triển khai chưa cần code chức năng)

(Các class này được tạo trong package repository)

**Task 3:** Xây dựng chức năng 1. Employee Management cho menu chính của chương trình.

*Gợi ý:*

1. Trong class EmployeeRepository sử dụng đọc ghi file để lưu trữ và khởi tạo dữ liệu. Viết các chức năng xử lý business cho phần hiển thị danh sách nhân viên, thêm mới nhân viên, chỉnh sửa thông tin cho nhân viên, xóa nhân viên và tìm kiếm theo tên nhân viên.

2. Tạo đối tượng EmployeeService trong FuramaController

3. Sử dụng các method business đã viết trong EmployeeService để hoàn thành chức năng Employee Management

4. Viết hàm kiểm tra dữ liệu đầu vào khi người dùng thực hiện thêm mới một nhân viên (add new Employee). Yêu cầu kiểm tra được hợp lệ của các thuộc tính: (**sử dụng regular expression, User Exception**)

* Mã nhân viên phải đúng định dạng: NV-YYYY, với YYYY là các số từ 0-9.

( “^NV-[0-9]{4}$” )

* Tên nhân viên phải viết hoa các ký tự đầu của mỗi từ.( “^[A-Z][a-z]\*(\s[A-Z][a-z]\*)\*$”)
* Phải đủ 18 tuổi (tính cả ngày + tháng)
* CMND phải đủ 9 hoặc 12 số. ( “^[0-9]{9}$ | ^[0-9]{12}”)
* Số điện thoại phải bắt đầu từ 0 và đủ 10 số.(“^0[0-9]{9}$”)
* Lương phải lớn hơn 0.

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

\* Lưu trữ và xử lý dữ liệu Employee thông qua file employee.csv được tạo trong thư mục data

**Lưu ý**: Xử lý Exception cho phần nhập dữ liệu từ người dùng, phần parse dữ liệu

**Task 4:** Xây dựng chức năng 2. Customer Management cho menu chính của chương trình.

*Gợi ý:*

1. Trong class CustomerRepository sử dụng đọc ghi file để lưu trữ và khởi tạo dữ liệu. Viết các chức năng xử lý business cho phần hiển thị danh sách khách hàng, thêm mới khách hàng, chỉnh sửa thông tin cho khách hàng, xóa khách hàng và tìm kiếm theo tên khách hàng.

2. Tạo đối tượng CustomerService trong FuramaController

3. Sử dụng các method business đã viết trong CustomerService để hoàn thành chức năng Customer Management

4. Viết hàm kiểm tra dữ liệu đầu vào khi người dùng thực hiện thêm mới một khách hàng (add new Customer). Yêu cầu kiểm tra được hợp lệ của các thuộc tính: (**sử dụng regular expression, User Exception**)

* Mã khách hàng phải đúng định dạng: KH-YYYY, với YYYY là các số từ 0-9.
* Tên khách hàng phải viết hoa các ký tự đầu của mỗi từ.
* Phải đủ 18 tuổi (tính cả ngày + tháng)
* CMND phải đủ 9 hoặc 12 số.
* Số điện thoại phải bắt đầu từ 0 và đủ 10 số.

\* Lưu trữ và xử lý dữ liệu Customer thông qua file customer.csv được tạo trong thư mục data

**Lưu ý**: Xử lý Exception cho phần nhập dữ liệu từ người dùng, phần parse dữ liệu

**Task 5:** Xây dựng chức năng 3. Facility Management cho menu chính của chương trình.

***Yêu cầu:*** *sử dụng LinkedHashMap*

*Gợi ý:*

1. Trong class FacilityRepository sử dụng LinkedHashMap *(sử dụng key là Facility, value là số lần đã sử dụng)*, static property, static block để lưu trữ và khởi tạo dữ liệu. Viết các chức năng xử lý business cho phần hiển thị danh sách facility, thêm mới facility, hiển thị danh sách facility cần bảo trì, xóa facility.

(mỗi khi khách hàng booking từ chức năng Booking thành công thì số lần sử dụng sẽ tăng 1 (task 6), hiện tại sau khi thêm mới thì thành công thì value nhận giá trị mặc định là 0)

2. Lưu ý khi người dùng chọn thêm mới dịch vụ sẽ hiển thị menu con cho phép người dùng chọn loại dịch vụ muốn thêm:

1. Add New Villa
2. Add New House
3. Add New Room
4. Back to menu

3. Tạo đối tượng FacilityService trong FuramaController

4. Sử dụng các method business đã viết FacilityService trong để hoàn thành chức năng Facility Management

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

* 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 30m2
* Chi phí thuê phải là số dương
* Số lượng người tối đa phải >0 và nhỏ hơn <20
* Số tầng phải là số nguyên dương.
* Kiểu thuê, Tiêu chuẩn phòng chuẩn hóa dữ liệu giống tên dịch vụ

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

**Lưu ý**: Xử lý Exception cho phần nhập dữ liệu từ người dùng, phần parse dữ liệu

**Task 6:** Xây dựng chức năng 4. Booking Management cho menu chính của chương trình.

***Yêu cầu:*** *Sử dụng static property, static block, comparable, comparator, TreeSet, Queue để thông tin booking không bị trùng lặp và hiển thị danh sách theo thứ tự ngày bắt đầu gần nhất, nếu ngày bắt đầu trùng nhau thì sắp xếp theo ngày kết thúc gần nhất*

1. Khi thực hiện booking phải hiển thị danh sách khách hàng cho nhân viên chọn mã khách hàng, hiển thị danh sách dịch vụ cho nhân viên chọn mã dịch vụ.

2. Danh sách booking hiển thị phải được sắp xếp theo thứ tự ngày tháng booking gần nhất, nếu trùng nhau thì sắp xếp theo ngày kết thúc thuê.

(update chức năng tính số lần sử dụng cho task 5 dựa vào danh sách booking trong tháng hiện tại, vì cuối tháng luôn luôn có bảo trì bảo dưỡng nên số lần sử dụng dụng sẽ được tính từ đầu tháng cho đến thời điểm hiện tại)

3. Khi tạo hợp đồng chỉ cho phép tạo hợp đồng theo thứ tự ngày bắt đầu booking sớm nhất được tạo hợp đồng trước, booking trễ hơn phải tạo hợp đồng sau.

*Gợi ý:*

1. Trong class BookingRepository sử dụng TreeSet, static property, static block để lưu trữ và khởi tạo dữ liệu. Viết các chức năng xử lý business cho phần hiển thị danh sách booking, thêm mới booking (Tạo class Booking Comparator implement Comparator để xử lý phần sắp xếp, override methodequals(), hashCode() để xử lý phần không trùng lặp).

2. Phần tạo hợp đồng chuyển danh sách booking từ Set sang Queue để xử lý booking theo thứ tự.

3. Tạo đối tượng BookingService trong FuramaController

4. Sử dụng các method business đã viết trong BookingService để hoàn thành chức năng Booking Managerment

**Task 7:** Xây dựng chức năng 5. Promotion Management cho menu chính của chương trình.

***Yêu cầu:*** *sử dụng TreeSet, Stack, static property, static block*

*Gợi ý:*

1. Trong class PromotionRepository sử dụng TreeSet, static property, static block để lưu trữ và khởi tạo dữ liệu.

2. Với chức năng hiển thị danh sách khách hàng đã sử dụng dịch vụ theo năm: dữ liệu được lấy từ danh sách booking, năm sử dụng dịch vụ cho phép người dùng nhập vào.

3. Với chức năng tặng voucher khuyến mãi. Cho phép người dùng nhập vào số lượng voucher khuyến mãi 10%, số lượng voucher khuyến mãi 20%, số lượng voucher khuyến mãi 50%, hệ thống sẽ lấy danh sách các khách hàng đã booking trong tháng hiện tại sắp xếp theo thứ tự booking và đẩy vào Stack (số lượng khách hàng = tổng số lượng voucher) sau đó tiến hành hiển thị danh sách khách hàng sẽ tặng voucher (các khách hàng đặt theo thứ tự muộn nhất sẽ nhận 10%, hết 10% mới sử dụng 20%, hết 20% mới được sử dụng 50%)

2. Tạo đối tượng PromotionService trong FuramaController

3. Sử dụng các method business đã viết trong PromotionService để hoàn thành chức năng Promotion Management

---------------------------------------------------HẾT------------------------------------------------------------