**I. Mô tả hệ thống .**

**1. Mô tả khái quát:**

-Hệ thống bán hàng BoEC gồm các mặt hàng được bán bao gồm: Sách, Điện tử và Thời trang.

-Đối tượng liên quan đến hệ thống bao gồm: Khách hàng, Nhân viên, Quản trị viên.

-Lưu trữ dữ liệu trong Mysql và SqlServer.

1.1. Khách hàng:

-Khách hàng có thể đặt hàng online hoặc mua trực tiếp.

-Khách hàng đặt online sẽ vào hệ thống, tìm kiếm sản phẩm cần mua trong 3 loại được bán, đưa sản phẩm vào giỏ hàng, nhập số lượng sản phẩm. Khi kết thúc quá trình tìm kiếm sản phẩm, khách hàng đặt hàng. Để xác đinh đặt hàng, khách hàng se nhập thông tin cá nhân vào hệ thống. Cửa hàng sẽ tự động gọi cho khách hàng qua số điện thoại cung cấp xác nhận mua hàng và chọn hình thức thanh toán.

- Khách hàng cũng có thể đăng ký thành viện để tích điểm và hưởng ưu đãi giảm giá.

1.2.Nhân viên liên quan đến quá trình mua bán:

-Nhân viên bao gồm Nhân viên Online, Nhân viên Quầy, Nhân viên kho.

-Nhân viên Online sẽ có nhiệm vụ tạo hóa đơn mua Online, và liên lạc với khách hàng.

-Nhân viện Quầy sẽ tạo hóa đơn trực tiếp ngay tại quầy cho khách hàng.

-Nhân viên Kho có nhiệm vụ cập nhật lượng hàng xuất nhập trong kho, báo cáo tồn kho...

+ Các nhân viên đều phải đăng nhập vào hệ thống khi thực hiện giao dịch.

1.3. Quản trị viên.

-Nhân viên quản trị phân chia quyền, thêm sửa xóa nhân viên, xem danh sách nhân viên và cập nhật thông tin về các đối tượng sử dụng hệ thống.

**2. Các chức năng chính trong hệ thống.**

-Quản lý hàng nhập.

+Thêm, sửa, xóa, tìm kiếm các mặt hàng mới và mặt hàng trong kho.

-Quản lý hàng bán.

+Thu thập các hóa đơn cập nhật lại số lượng hàng trong kho.

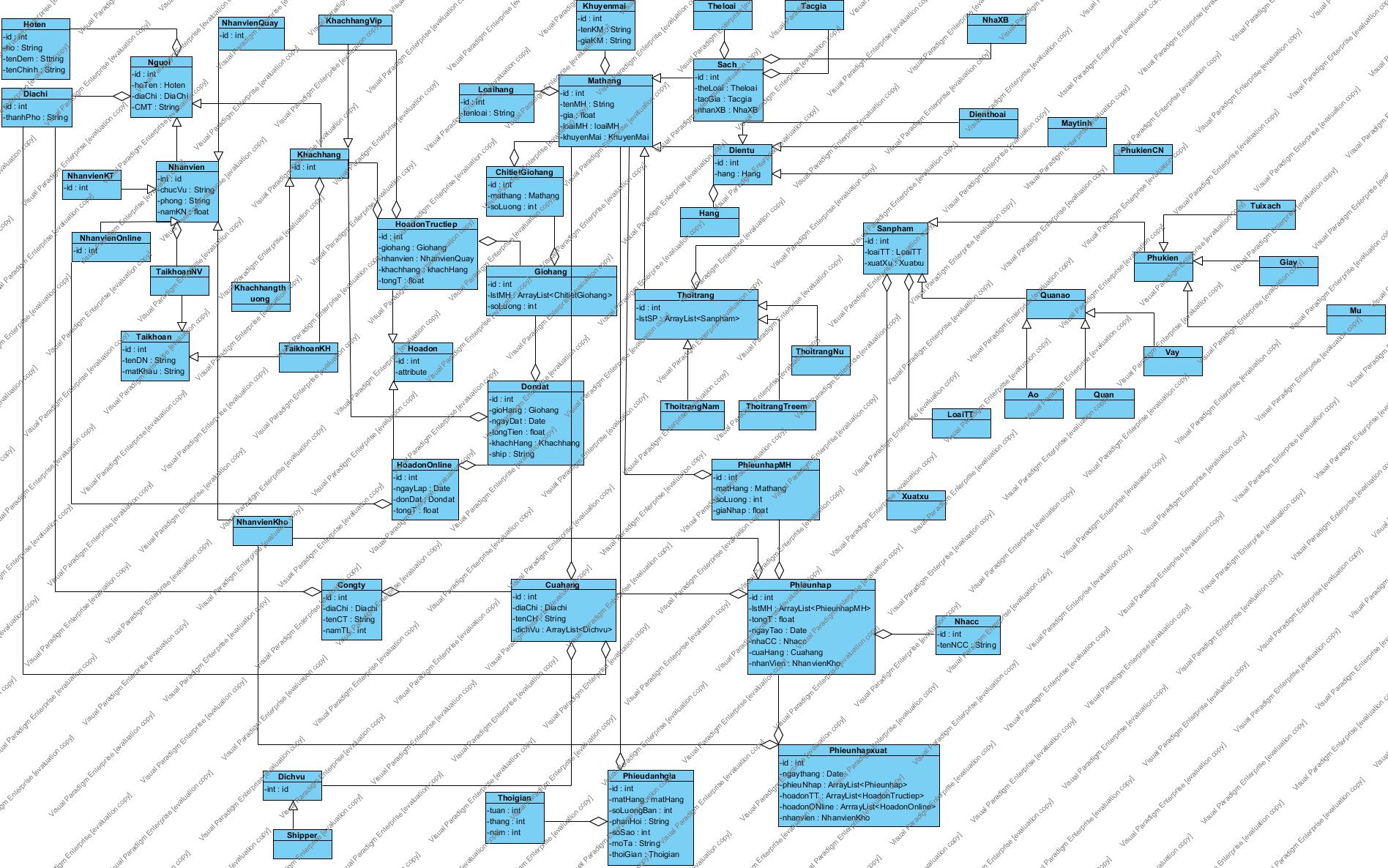
-Quản lý nhân sự, báo cáo, thống kê:

+Thêm, sửa, xóa, tìm kiếm thông tin va lương nhân viên, khách hàng.

+Tạo báo cáo thống kê lượng hàng bán theo kỳ, hàng tồn kho...

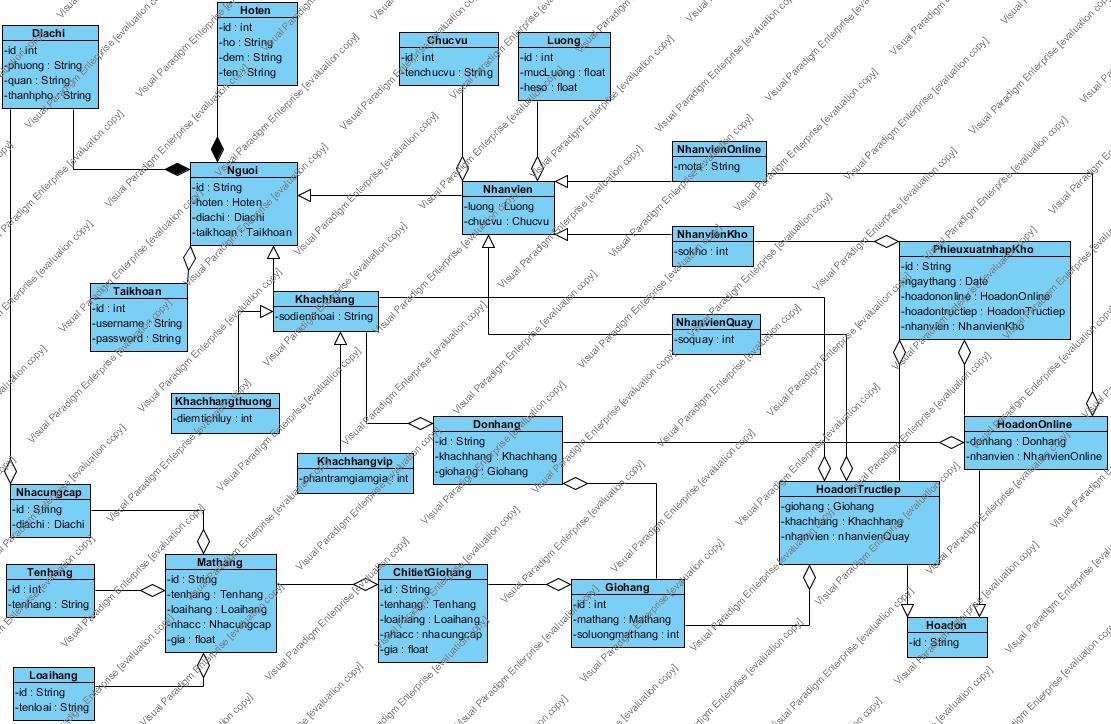
-Quản lý đặt hàng.

**II. Biểu đồ lớp thiết kế.(>50 lớp)**



**III> Tìm hiểu và áp dụng pattern.**

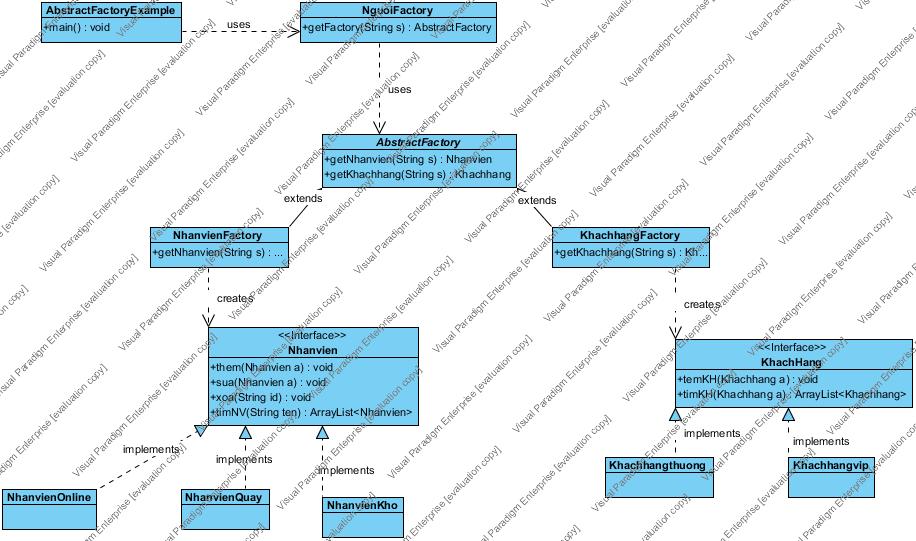
**1. Biểu đồ lớp áp dụng:**



**2. Pattern AbstractFactory.**

- Pattern này được sử dụng khi hệ thống có nhiều thuộc tính, hàng vi tương tự nhau. Cung cấp một lớp trừu tượng có chức năng tạo ra một tập hợp các đối tượng liên quan hoặc phụ thuộc lẫn nhau mà không chỉ ra đó là lớp cụ thể nào tại thời điểm thiết kế.

-Sơ đồ:



Bước 1: Tạo lớp interface: Khachhang.java

Tạo lớp Khachhangthuong implements lớp Khachhang

Tạo lớp Khachhangvip implements lớp Khachhang

Bước 2: Tạo lớp interface: Nhanvien.java

Tạo lớp NhanvienKho implement lớp Nhanvien

Tạo lớp NhanvienQuay implements lớp Nhanvien

Tạo lớp NhanvienOnline implements lớp Nhanvien

Bước 3: Tạo lớp Abstracts: AbstractFactory

Bước 4: Tạo lớp KhachhangFactory extends AbstractFactory

Bước 5: Tạo lớp NhanvienFactory extends AbstractFactory

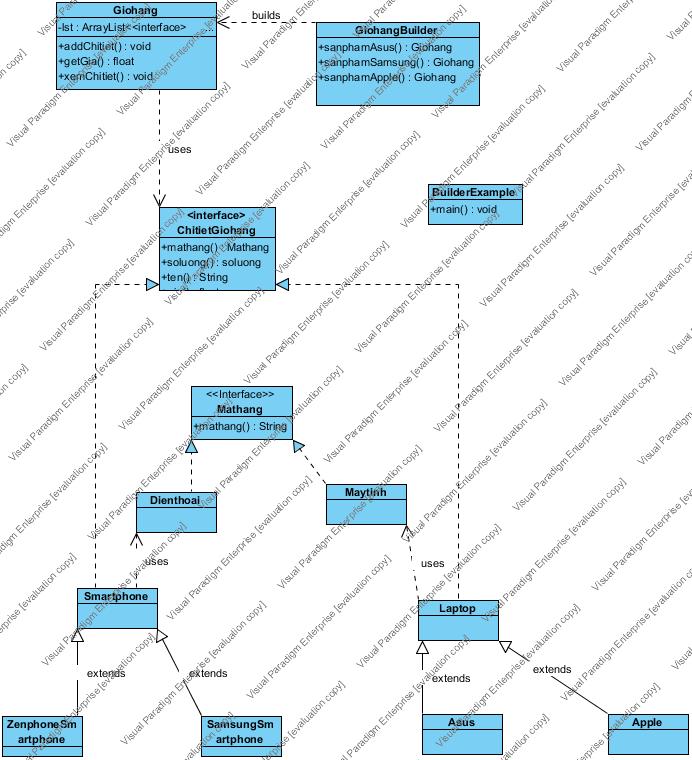
Bước 6: Tạo lớp FactoryNguoi

Bước 7: Tạo lớp FactoryPatternExample.

**2. Buider pattern**

-Mục đích tách rời việc tạo một đối tượng phức tạp khỏi biểu diễn của nó sao cho cùng một tiến trình xây dựng có thể tạo được các biểu diễn khác nhau.

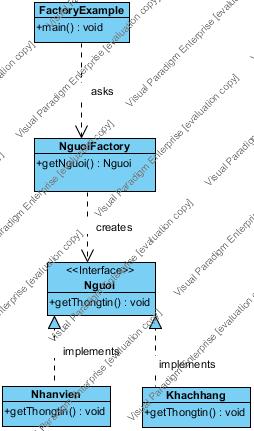
-Sơ đồ:

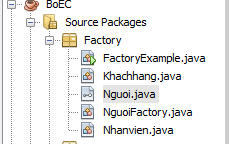


**10. Factory pattern:**

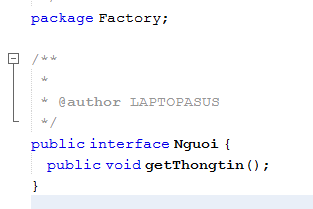
- Nhằm giải quyết vấn đề tạo ra một đối tượng mà không cần thiết chỉ ra một cách chính xác lớp nào sẽ được tạo ra. Factory pattern giải quyết vấn đề này bằng cách định nghĩa một phương thức cho việc tạo đối tượng và các lớp con kế thừa có thể override để chỉ rõ đối tượng nào sẽ được tạo.

-Sơ đồ:

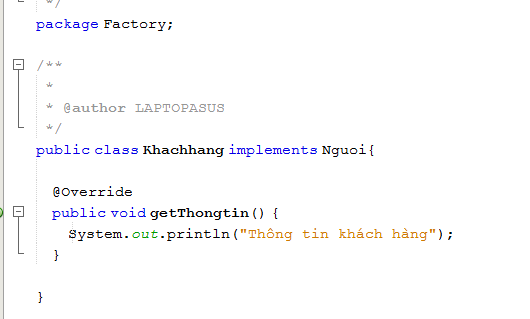




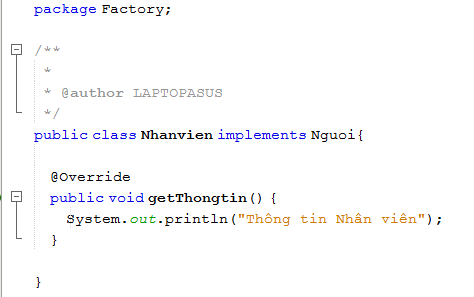
***Bước 1: Tạo Interface: Nguoi.java***



***Bước 2: Tạo lớp Khachhang implements lớp Nguoi***



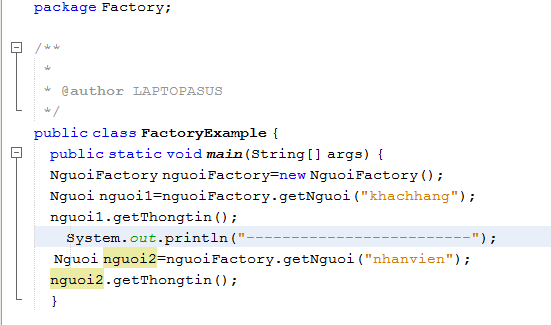
***Tạo lớp Nhanvien implements lớp Nguoi :***



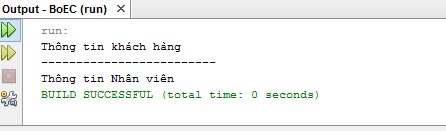
***Bước 3 : Tạo lớp NguoiFactory***



***Bước 4 : Tạo lớp FactoryPatternExample***



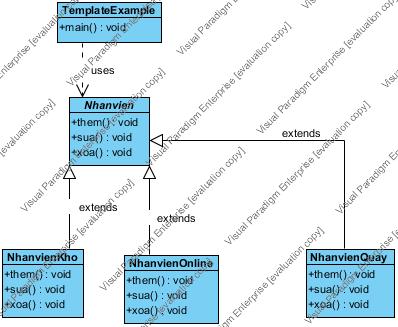
***Kết quả :***

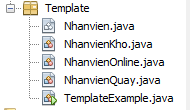


**11. Template pattern**

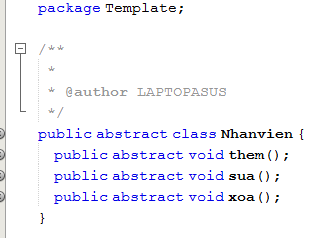
- Định nghĩa phần khung của một thuật toán, tức là một thuật toán tổng quát gọi đến một số phương thức chưa được cài đặt trong lớp cơ sở; việc cài đặt các phương thức được ủy nhiệm cho các lớp kế thừa

-Sơ đồ.

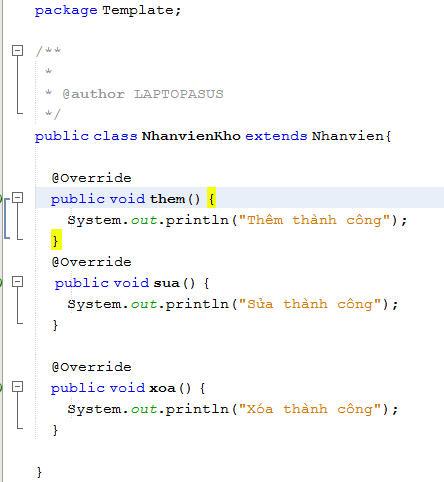


****

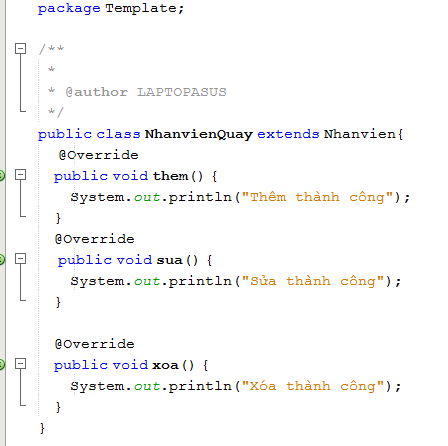
***Bước 1: Tạo 1 abstract class: Nhanvien***



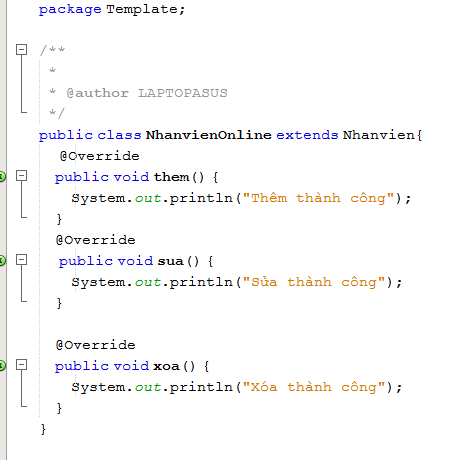
***Bước 2:Tạo class NhanvienKho extends Nhanvien***

******

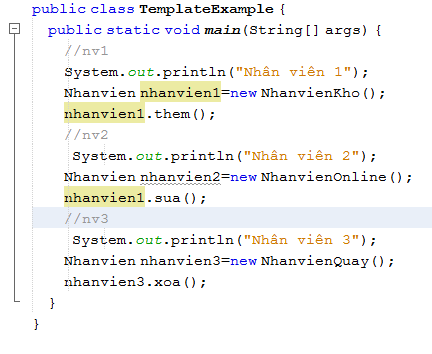
***Bước 3: Tạo class NhanvienQuay extends Nhanvien***



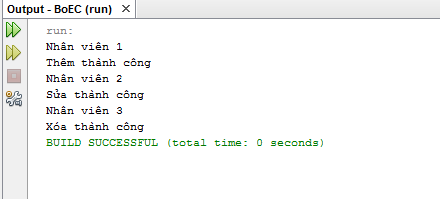
***Bước 4:Tạo class NhanvienOnline extends Nhanvien***



***Bước 5: Tạo class TemplateExample***



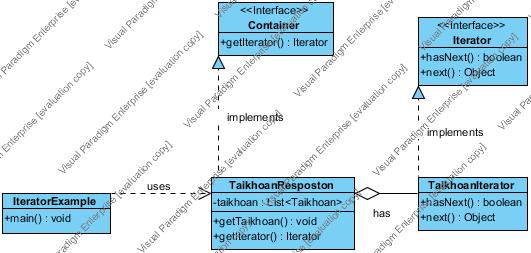
***Kết quả:***

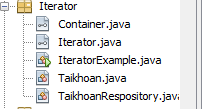


**12. Iterator pattern**

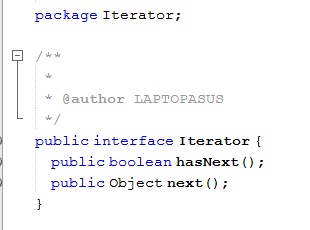
- Truy xuất các phần tử của đối tượng dạng tập hợp tuần tự (list, array, …) mà không phụ thuộc vào biểu diễn bên trong của các phần tử.

- Sơ đồ:

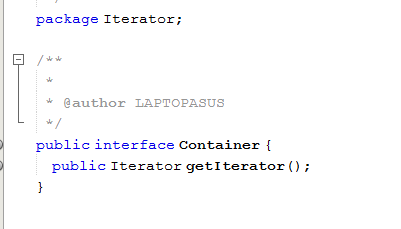




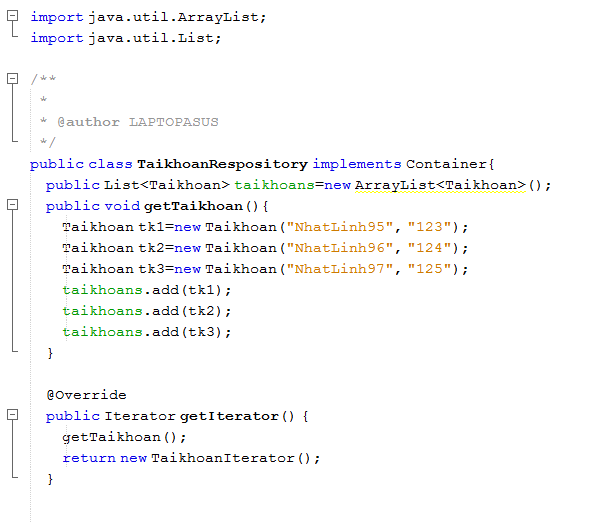
***Bước 1: Tạo interface Iterator***

******

***Bước 2: Tạo interface Container***



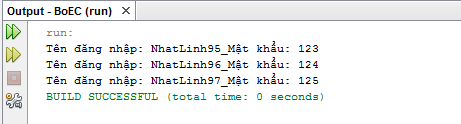
***Bước 3: Tạo class TaikhoanRespository implements Container***

***Bước 4: Tạo class IteratorExample***



***Kết quả:***



**VI. Code một số chức năng theo MVC.**