**TRƯỜNG ĐẠI HOC BÌNH DƯƠNG**

**KHOA CÔNG NGHỆ THÔNG TIN, ROBOT**

**VÀ TRÍ TUỆ NHÂN TẠO**

---**oOo**---



**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Tên đề tài:**

**LẬP TRÌNH QUẢN LÝ HOẠT ĐỘNG MUA BÁN Ở MỘT SIÊU THỊ**

Giảng viên hướng dẫn: **Thầy Huỳnh Anh Tuấn**

**Nhóm tiểu luận: 2.1.2**

Sinh viên thực hiện:

**Lê Tuấn Hải: 23050116**

**Nguyễn Minh Cang: 23050120**

**Nguyễn Tuấn Anh: 23050118**

**Lê Ngọc Sang: 23050084**

**Bình Dương, tháng 05, năm 2024**

**LỜI CAM ĐOAN**

Nhóm sinh viên ký tên dưới đây xin cam đoan: bài tiểu luận này là kết quả nghiên cứu, phát triển của cá nhân (hoặc nhóm), không phải là sản phẩm được sao chép hoặc đã được tác giả khác công bố.

**MỤC LỤC**

[I. Phần giới thiệu 3](#_Toc164962482)

[1. Thông tin chung 3](#_Toc164962483)

[2. Công cụ thực hiện 3](#_Toc164962484)

[3. Kế hoạch thực hiện 3](#_Toc164962485)

[II. phát biểu bài toán 3](#_Toc164962486)

[1. Hiện trạng 3](#_Toc164962487)

[2. Các yêu cầu chức năng của chương trình 3](#_Toc164962488)

[III. mô hình hóa các đối tượng 4](#_Toc164962489)

[1. Sơ đồ lớp thực thể 4](#_Toc164962490)

[2. Xác định class trung tâm 4](#_Toc164962491)

[IV. Kết quả lập trình 4](#_Toc164962492)

[1. Source code 4](#_Toc164962493)

[2. Thử nghiệm (Testing) 4](#_Toc164962494)

[V. Kết Luận – Kiến nghị 4](#_Toc164962495)

[1. Đánh giá kết quả đạt được của đề tài so với yêu cầu của đề tài 4](#_Toc164962496)

[2. Đề xuất xu hướng phát triển của đề tài 4](#_Toc164962497)

[DANH MỤC TÀI LIỆU THAM KHẢO 5](#_Toc164962498)

# Phần giới thiệu

## Thông tin chung

* Tên đề tài: Lập trình quản lý hoạt động mua bán ở một siêu thị
* Danh sách thành viên của nhóm:
* Lê Tuấn Hải
* Nguyễn Minh Cang
* Nguyễn Tuấn Anh
* Lê Ngọc Sang
* Ngày hoàn thành: 15/5/2024

## Công cụ thực hiện

* Công cụ mô hình hóa: IBM Rational Rose 7.0
* Công cụ lập trình: MS Visual Studio 2022

## Kế hoạch thực hiện

Bảng phân công các thành viên của nhóm và tiến độ hoàn thành từng công việc:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ tên và mã sinh viên | Công việc được phân công | Tỷ lệ hoàn thành công việc(%) |
| 1 | Lê Tuấn Hải:23050116 | Viết Word,làm code,sửa code | 100% |
| 2 | Nguyễn Minh Cang: 23050120 | Làm code,ý tưởng word | 100% |
| 3 | Nguyễn Tuấn Anh: 23050118 | Làm code, sửa code,ý tưởng ibm | 100% |
| 4 | Lê Ngọc Sang: 23050084 | Vẽ IBM , ý tưởng word, làm code | 100% |

# phát biểu bài toán

## Hiện trạng

Đây là chương trình viết về  quản lý hoạt động mua bán ở một siêu thị. Chương trình gồm có các đối tượng sau: Nhân viên, Khách hàng, Hàng hóa, Hóa đơn, Chi tiết hóa đơn, Ca làm:

-Nhân viên:

* Không cho phép nhập “Họ tên nhân viên” là chuỗi rỗng hoặc khoảng trắng và ký tự đặc biệt
* Không cho phép nhập “Mã số nhân viên” là chuỗi rỗng, khác 8 ký tự và kí tự đặc biệt
* Không cho phép nhập “địa chỉ’ là chuỗi rỗng hoặc khoảng trắng và kí tự đặc
* Không cho phép nhập “ngày sinh” nhân viên bé hơn 18 và nhập theo quy tắc (dd/MM/yyyy) hoặc (yyyy, MM, dd)
* “Giới tính” nhận 1 trong 2 giá trị: 0 - Nam hoặc 1 - Nữ
* “Số điện thoại” phải có đúng 10 ký tự số, bắt đầu bằng “0” và không chứa kí tự đặc biệt
* Không cho phép trong danh sách có 2 nhân viên trùng nhau về: Mã số nhân viên, số điện thoại
* Không cho phép nhập “chức vụ” là chuỗi rỗng hoặc khoảng trắng và kí tự đặc biệt
* “Lương” của nhân viên lớn hơn hoặc bằng 0
* Không cho phép nhập “Phụ cấp” là chuỗi rỗng hoặc khoảng trắng và kí tự đặc biệt
* Cuối cùng, tạo một đối tượng NhanVien mới với các thông tin đã nhập và thêm đối tượng đó vào danh sách nhân viên (DSNV).
* Khách hàng:
* Không cho phép nhập “Họ tên khách hàng” là chuỗi rỗng hoặc khoảng trắng và kí tự đặc biệt
* Không cho phép “Mã khách hàng” khác 7 ký tự và kí tự đặc biệt
* Không cho phép nhập “Địa chỉ’ là chuỗi rỗng hoặc khoảng trắng và kí tự đặc biệt
* Không cho phép nhập “Ngày sinh” bé hơn 18 và nhập theo quy tắc (dd/MM/yyyy) hoặc (yyyy, MM, dd)
* “Giới tính” nhận 1 trong 2 giá trị: 0 - Nam hoặc 1 - Nữ
* Không cho phép “Điểm tích lũy” bé hơn “0” và lớn lơn “1000”
* Không cho phép trong danh sách có 2 khách hàng trùng nhau về: Mã số khách hàng, số điện thoại
* Cuối cùng, tạo một đối tượng KhachHang mới với các thông tin đã nhập và thêm đối tượng đó vào danh sách khách hàng (DSKH)
* Ca Làm:
* Không cho phép nhập “ID ca làm” là chuỗi rỗng, khác 6 kí tự và kí tự đặc biệt
* Không cho phép nhập “tên ca” là chuỗi rỗng và kí tự đặc biệt
* Không cho phép “Ngày làm” phải là ngày làm bé hơn ngày hiện và tại nhập theo quy tắc (dd/MM/yyyy) hoặc (yyyy, MM, dd)
* “Giờ làm” và “giờ nghĩ giữa ca” là phải cùng kiểu dữ liệu và cùng một loạt điều kiện. Giá trị nhập được gán phải nằm trong khoảng từ 00:00:00 đến 23:59:59, nằm trong phạm vi 24 giờ của một ngày và nhập theo quy tắc HH:mm
* Không cho phép nhập “Mã số nhân viên” là chuỗi rỗng, khác 8 kí tự và kí tự đặc biệt
* Cuối cùng, tạo một đối tượng CaLam mới với các thông tin đã nhập và thêm đối tượng đó vào danh sách ca làm (DSCL)
* Hóa Đơn:
* Không cho phép nhập “ Mã hóa đơn” là rỗng hoặc khoảng trắng và ký tự đặc biệt.
* “ Ngày lập hóa đơn” phải theo định dạng "Ngày/tháng/năm" (dd/MM/yyyy)
* Không cho phép nhập “mã khách hàng” là chuỗi rỗng hoặc khoảng trắng và ký tự đặc biệt.
* Không cho phép nhập “số điện thoại cửa hàng” là chuỗi rỗng hoặc khoảng trắng, khác 10 kí tự và ký tự đặc biệt.
* Không cho phép nhập “ mã nhân viên lập hóa đơn” là trống hoặc khoảng trắng và tìm kiếm trong danh sách nhân viên (DSNV) để tìm nhân viên có mã trùng khớp với mã nhập vào.
* Không cho phép nhập “địa chỉ khách hàng” là trống hoặc khoảng trắng và tìm kiếm trong danh sách khách hàng (DSKH) để tìm khách hàng có địa chỉ trùng khớp với địa chỉ nhập vào
* Không cho phép nhập “Tổng tiền hóa đơn ” là trống hoặc khoảng trắng, bé hơn 0 và kí tự đặ biệt
* Cuối cùng, tạo một đối tượng HoaDon mới với các thông tin đã nhập và thêm đối tượng đó vào danh sách hóa đơn (DSHD).
* Hàng Hóa:
* Không cho phép nhập “Tên hàng hóa” là chuỗi rỗng hoặc khoảng trắng và kí tự đặc biệt
* Không cho phép nhập “Mã hàng hóa” là chuỗi rỗng hoặc khoảng trắng và kí tự đặc biệt
* Không cho phép nhập “Nơi sản xuất” là chuỗi rỗng hoặc khoảng trắng và kí tự đặc biệt
* Bắt buộc phải nhập “Đơn vị tính”, và “Đơn vị tính” phải là một số nguyen
* Không cho phép nhập “Loại hàng hóa” là chuỗi rộng hoặc khoảng trắng và kí tự đặc biệt
* Bắt buộc phải nhập “Giá bán”, và “Giá bán” của hàng hóa phải là một số dương
* Không cho phép nhập “Hạn sử dụng” là thời điểm trước thời điểm hiện tại, và nhập theo quy tắc (dd/MM/yyyy).
* Cuối cùng, tạo một đối tượng HangHoa mới với các thông tin đã nhập và thêm đối tượng đó vào danh sách hàng hóa (DSHD).
* Chi tiết hóa đơn :
* Không cho phép nhập “ Mã hóa đơn” là rỗng hoặc khoảng trắng và ký tự đặc biệt, dựa trên class hóa đơn.
* Phải tìm hóa đơn trong danh sách DSHD dựa trên mã hóa đơn.
* Phải kiểm tra xem hóa đơn này đã được thêm chi tiết hay chưa.
* Mã hàng hóa phải tồn tại trong danh sách DSHH và hiển thị thông tin hàng hóa và yêu cầu người dùng phải nhập số lượng.
* Kiểm tra số lượng phải là số thực lớn hơn không.
* Tạo đối tượng ChiTietHoaDon mới và thêm vào danh sách DSCTHD và cập nhật tổng số tiền (total).
* Người dùng nhập "exit", tổng số tiền phải khớp với TT\_new. Nếu khớp, thêm hóa đơn vào danh sách Da\_Nhap\_CT và hiển thị thông báo "Thêm chi tiết vào HD thành công".
* Cuối cùng, tạo một đối tượng HangHoa mới với các thông tin đã nhập và thêm đối tượng đó vào danh sách hàng hóa (DSHH).

## Các yêu cầu chức năng của chương trình

Các chức năng chính của chương trình bao gồm:

(Nhập và thêm – Cập nhật– Hiển thị–Xóa-In)

* Tạo menu chính khi chương trình mới khởi động
* Quản lý nhân viên:
* Nhập thông tin và thêm nhân viên mới vào danh sách
* Cập nhật nhân viên theo mã nhân viên
* Hiển thị toàn bộ danh sách nhân viên đã thêm
* Xóa thông tin nhân viên đã thêm
* In danh sách nhân viên bằng mã nhân viên
* Quản lý khách hàng:
* Nhập thông tin và thêm khách mới vào danh sách
* Cập nhật khách hàng theo mã khách hàng
* Hiển thị toàn bộ danh sách khách hàng đã thêm
* Xóa thông tin khách hàng đã thêm
* In danh sách khách hàng bằng mã khách hàng
* Quản lý ca làm:
* Nhập thông tin và thêm ca làm mới vào danh sách
* Cập nhật ca làm theo ID ca
* Hiển thị toàn bộ danh sách ca làm đã thêm
* Xóa thông tin ca làm đã thêm
* In danh sách ca làm bằng ID ca làm
* Quản lý hóa đơn:
* Nhập thông tin và thêm hóa đơn mới vào danh sách
* Hiển thị toàn bộ danh sách hóa đơn đã thêm
* Xóa thông tin hóa đơn đã thêm
* In mã danh sách hóa đơn bằng mã hóa đơn
* Quản lý hàng hóa:
* Nhập thông tin và thêm hàng hóa mới vào danh sách
* Cập nhật hàng hóa theo mã hàng hóa
* Hiển thị toàn bộ danh sách hàng hóa làm đã thêm
* Xóa thông tin hàng hóa đã thêm
* In danh sách hàng hóa bằng mã hàng hóa
* Quản lý chi tiết hóa đơn:
* Nhập thông tin và thêm chi tiết hóa đơn mới vào danh sách
* Hiển thị toàn bộ danh sách ca làm đã thêm
* Thoát chương trình

# mô hình hóa các đối tượng

## Sơ đồ lớp thực thể

Căn cứ vào nội dung mô tả trong đề tài “Lập trình quản lý hoạt động mua bán của một siêu thị”, nhóm chúng em chọn lọc được 6 đối tượng thực thể ( Nhân Viên, Khách Hàng, Ca Làm, Hàng Hóa, Hóa Đơn, Chi Tiết Hóa Đơn, Thuốc) để trừu tượng hóa thành các class.

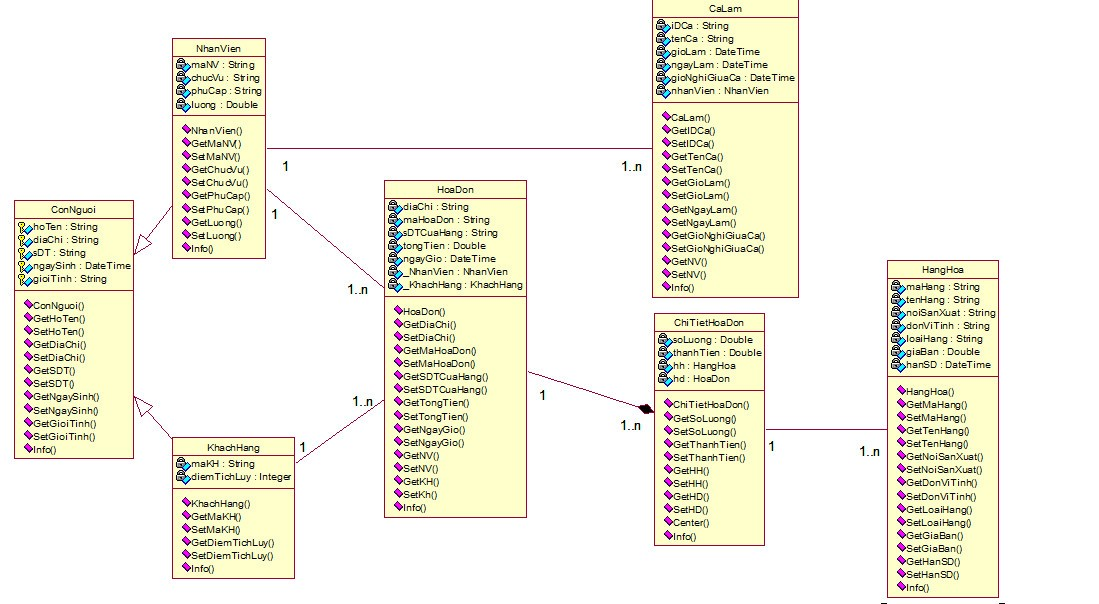
Mỗi class gồm tên class và các thuộc tính (attribute):

* Class “Nhân Viên”
* Họ và tên: được viết theo dạng chuỗi (String)
* Mã số: được viết theo dạng chuỗi (String)
* Địa chỉ: được viết theo dạng chuỗi (String)
* Ngày sinh: được viết theo dạng ngày tháng (DateTime)
* Giới tính : được viết theo dạng chuỗi (String)
* Mã nhân viên: được viết theo dạng chuỗi (String)
* Chức vụ: được viết theo dạng chuỗi (String)
* Phụ cấp: được viết theo dạng chuỗi (String)
* Lương: được viết theo dạng số thực (Double)
* Class “Khách Hàng”
* Họ và tên: được viết theo dạng chuỗi (String)
* Mã số: được viết theo dạng chuỗi (String)
* Địa chỉ: được viết theo dạng chuỗi (String)
* Ngày sinh: được viết theo dạng ngày tháng (Date)
* Giới tính: được viết theo dạng chuỗi (String)
* Mã khách hàng: được viết theo dạng chuỗi (String)
* Điểm tích lũy: được viết theo dạng chuỗi (String)
* Class “Ca Làm”
* iD ca: được viết theo dạng chuỗi (String)
* Tên ca: được viết theo dạng chuỗi (String)
* Ngày làm: được viết theo dạng ngày tháng (DateTime)
* Giờ làm: được viết theo dạng ngày tháng (DateTime)
* Giờ nghỉ giữa ca: được viết theo dạng ngày tháng (DateTime)
* Mã nhân viên: được viết theo dạng chuỗi (String)
* Class “Hóa Đơn”
* Địa chỉ: được viết theo dạng chuỗi (String)
* Mã hóa đơn: được viết theo dạng chuỗi (String)
* Số điện thoại cửa hàng: được viết theo dạng chuỗi (String)
* Tổng tiền: được viết theo dạng số thực (Double)
* Ngày giờ: được viết theo dạng ngày tháng (DateTime)
* Mã nhân viên: được viết theo dạng chuỗi (String)
* Mã khách hàng: được viết theo dạng chuỗi (String)
* Class “Hàng Hóa”
* Mã hàng: được viết theo dạng chuỗi (String)
* Tên hàng: được viết theo dạng chuỗi (String)
* Nơi sản xuất: được viết theo dạng chuỗi (String)
* Đơn vị tính: được viết theo dạng chuỗi (String)
* Loại hàng: được viết theo dạng chuỗi (String)
* Giá bán: được viết theo dạng số thực (Double)
* Hạn sử dụng: được viết theo dạng ngày tháng (DateTime)

Các mối quan hệ giữa các class:

* Vì 2 class “Nhân viên”, “Khách hàng”có một số thuộc tính (attribute) và đặc điểm chung:
* Họ và tên: được viết theo dạng chuỗi (String)
* Mã số: được viết theo dạng chuỗi (String)
* Địa chỉ: được viết theo dạng chuỗi (String)
* Ngày sinh: được viết theo dạng ngày tháng (Date)
* Giới tính: được viết theo dạng đúng hoặc sai (Bool)
* Áp dụng phép generalization (kế thừa). Tạo một class “Con Người” là class cha chứa các thuộc tính (attribute) chung của 2 class “Nhân viên”, “Khách hàng”. Các thuộc tính chung này được khai báo dưới dạng (protected) cho phép các class con là 2 class “Nhân viên”, “Khách hàng” truy cập. Không cần khai báo lại các thuộc tính chung từ class cha vào các class con.
* Mối quan hệ kết hợp (Association) là mối quan hệ giữa các class khác nhau, cho phép các class tương tác với nhau để thực hiện một hoạt động cụ thể.Mang tính đa dạng, mối quan hệ kết hợp có thể là một-một, một-nhiều, nhiều-nhiều, tùy thuộc vào số lượng đối tượng tương tác với nhau.
* Khách hàng khi mua hàng sẽ được nhận lại hóa đơn và cả hai không có những thuộc tính, đặc điểm chung nên hai class “Khách hàng” và “Hóa đơn” là mối quan hệ kết hợp (Association). Một khách hàng có thể nhận được một hoặc nhiều hóa đơn, nhưng một hóa đơn chỉ thuộc về một khách hàng
* Nhân viên sẽ in ra hóa đơn và cả hai không có những thuộc tính, đặc điểm chung nên hai class “Nhân viên” và class “Hóa đơn” là mối quan hệ kết hợp (Association). Một nhân viên có thể tạo ra 0 hoặc nhiều hóa đơn, một hóa đơn chỉ có một nhân viên tạo ra nó.
* Vì class “Nhân viên” và “khách hàng” đều liên kết đến class “Hóa đơn”. Ta có thể nói khi khách hàng mua hàng, thì khách hàng sẽ tiếp xúc với nhân viên để mua và nhận hóa đơn.
* Hàng hóa được mua sẽ được ghi lại chi tiết trong chi tiết hóa đơn và cả hai không có những thuộc tính, đặc điểm chung nên hai class “Hàng hóa” và “Chi tiết hóa đơn” là mối quan hệ kết hợp (Association). Hàng hóa có thể có một hoặc nhiều trong một chi tiết hóa đơn, và một chi tiết hóa đơn có thể dùng cho một hoặc nhiều hàng hóa.
* Trong một hóa đơn sẽ có tồn tại chi tiết hóa đơn, mà chi tiết hóa đơn lại có mối quan hệ phụ thuộc vào hóa đơn, vì khi không còn hóa đơn thì chi tiết hóa đơn sẽ mất, nên hai class “Hóa đơn” và “Chi tiết hóa đơn” có mối quan hệ “Composition”. Một hóa đơn chỉ có một chi tiết hóa đơn, nhưng một chi tiết hóa đơn có thể dùng cho một hoặc nhiều hóa đơn.
* Vì class “Hóa đơn” cần “Chi tiết hóa đơn”, mà từ “Chi tiết hóa đơn” ta có thể tìm kiếm “Hàng hóa”, nên ta có thể nói một hóa đơn cần chi tiết hóa đơn để ta có thể hiểu rõ được các thông tin cần thiết đối với hàng hóa mà ta mua. (Class “Hóa đơn” liên kết class “Chi tiết hóa đơn”, và từ class “Chi tiết hóa đơn” lại liên kết với “Hàng hóa”).
* Khi nhân viên đi làm việc phải theo ca làm và cả hai không có những thuộc tính, đặc điểm chung nên hai class “Nhân viên” và class “Ca làm” là mối quan hệ kết hợp (Association). Một nhân viên có thể có một hoặc nhiều ca làm, một ca làm có thể có 1 hoặc nhiều nhân viên.
* Vì “Nhân viên” liên kết với “Ca làm” nên ta có thể nói, nhân viên sẽ làm việc theo ca của bản thân, và từ trong ca làm này nhân viên mới có thể liên kết với nhều mối quan hệ khác.

Sơ đồ các lớp thực thể (Entity Class Diatram) theo kết quả phân tích ở trên được vẽ bằng công cụ (IBM Rational Rose):



## Xác định class trung tâm

Class trung tâm của đề tài là: Hóa đơn

# Kết quả lập trình

## Source code

Class ConNguoi:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tieu\_Luan\_2.\_1.\_2

{

internal class ConNguoi

{

protected string hoTen, diaChi, gioiTinh, sDT;

protected DateTime ngaySinh;

public ConNguoi(string ht, string dc, string gt, string sdt, DateTime ns)

{

this.hoTen = ht; this.diaChi = dc; this.gioiTinh = gt; this.sDT = sdt; this.ngaySinh = ns;

}

public string HoTen

{

get { return hoTen; }

set { { hoTen = value; } }

}

public string SDT

{

get { return sDT; }

set { if (sDT.Length == 10 && long.TryParse(sDT, out long x)) { sDT = value; } }

}

public DateTime NgaySinh

{

get { return ngaySinh; }

set { if (DateTime.Now.Year - ngaySinh.Year >= 6) { ngaySinh = value; } }

}

public string DiaChi

{

get { return diaChi; }

set

{

if (!string.IsNullOrEmpty(value))

{

DiaChi = value;

}

}

}

public string GioiTinh

{

get { return gioiTinh; }

set { gioiTinh = value; }

}

public virtual void Info()

{

Console.WriteLine($"Tên: {HoTen}");

Console.WriteLine($"Ngày sinh: {NgaySinh.Day}/{NgaySinh.Month}/{NgaySinh.Year}");

Console.WriteLine($"Giới Tính: {GioiTinh}");

Console.WriteLine($"Địa chỉ: {DiaChi}");

Console.WriteLine($"Số điện thoại: {SDT}");

}

}

}

Class NhanVien:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tieu\_Luan\_2.\_1.\_2

{

internal class NhanVien : ConNguoi

{

private string maNV, chucVu, phuCap;

private double luong;

public NhanVien(string ht, string dc, string gt, string sdt, DateTime ns, string maNV, string chucVu, string phuCap, double luong) : base(ht, dc, gt, sdt, ns)

{

this.maNV = maNV;

this.chucVu = chucVu;

this.phuCap = phuCap;

this.luong = luong;

}

public string MaNV

{

get { return maNV; }

set { maNV = value; }

}

public string ChucVu

{

get { return chucVu; }

set { chucVu = value; }

}

public string PhuCap

{

get { return phuCap; }

set

{

if (string.IsNullOrWhiteSpace(phuCap))

phuCap = value;

}

}

public double Luong

{

get { return luong; }

set { luong = value; }

}

public override void Info()

{

base.Info();

Console.WriteLine($"Mã nhân viên: {MaNV} ");

Console.WriteLine($"Chức vụ: {ChucVu}");

Console.WriteLine($"Luơng : {Luong}");

Console.WriteLine($"Phụ cấp : {PhuCap}");

}

}

}

Class KhachHang:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tieu\_Luan\_2.\_1.\_2

{

internal class KhachHang : ConNguoi

{

private string maKH; private int diemTichLuy;

public KhachHang(string ht, string dc, string gt, string sdt, DateTime ns, string mkh, int dtl) : base(ht, dc, gt, sdt, ns)

{

maKH = mkh;

diemTichLuy = dtl;

}

public int DiemTichLuy => diemTichLuy;

public string MaKH

{

get { return maKH; }

set { maKH = value; }

}

public override void Info()

{

base.Info();

Console.WriteLine($"Mã khách hàng: {MaKH}");

Console.WriteLine($"Điểm tích lũy: {DiemTichLuy}");

}

}

}

Class CaLam:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tieu\_Luan\_2.\_1.\_2

{

internal class CaLam

{

private string iDCa, tenCa;

private DateTime ngayLam, gioLam, gioNghiGiuaCa;

private NhanVien nhanVien;

public NhanVien NV => nhanVien;

public CaLam(NhanVien nhanVien, string iDCa, string tenCa, DateTime ngayLam, DateTime gioLam, DateTime gioNghiGiuaCa)

{

this.nhanVien = nhanVien;

this.iDCa = iDCa;

this.tenCa = tenCa;

this.NgayLam = ngayLam;

this.GioLam = gioLam;

this.GioNghiGiuaCa = gioNghiGiuaCa;

}

public string IDCa { get => iDCa; set { iDCa = value; } }

public string TenCa { get => tenCa; set { tenCa = value; } }

public DateTime NgayLam

{

get => ngayLam;

set { ngayLam = value; }

}

public DateTime GioLam

{

get => gioLam;

set

{

if (value.TimeOfDay >= new TimeSpan(0, 0, 0) && value.TimeOfDay <= new TimeSpan(23, 59, 59))

gioLam = value;

}

}

public DateTime GioNghiGiuaCa

{

get => gioNghiGiuaCa;

set

{

if (value.TimeOfDay >= new TimeSpan(0, 0, 0) && value.TimeOfDay <= new TimeSpan(23, 59, 59))

gioNghiGiuaCa = value;

}

}

public void Info()

{

Console.WriteLine($"ID ca làm: {IDCa} ");

Console.WriteLine($"Tên ca làm: {TenCa}");

Console.WriteLine($"Ngày làm: {NgayLam.Day}/{NgayLam.Month}/{NgayLam.Year}");

Console.WriteLine($"Giờ làm: {GioLam.Hour}:{GioLam.Minute}");

Console.WriteLine($"Giờ nghỉ giữa ca : {GioNghiGiuaCa.Hour}:{GioNghiGiuaCa.Minute}");

Console.WriteLine($"Mã nhân viên: {NV.MaNV}");

}

}

}

Class HangHoa:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tieu\_Luan\_2.\_1.\_2

{

internal class HangHoa

{

protected string tenHang, maHang, noiSanXuat, donViTinh, loaiHang;

protected double giaBan;

protected DateTime hanSD;

public HangHoa(string th, string mh, string nsx, string dvt, string lh, double gb, DateTime hsd)

{

this.tenHang = th;

this.maHang = mh;

this.noiSanXuat = nsx;

this.donViTinh = dvt;

this.loaiHang = lh;

this.giaBan = gb;

this.hanSD = hsd;

}

public string MaHang => maHang;

public string TenHang

{

get { return tenHang; }

set

{

if (tenHang.Length != 0)

{ tenHang = value; }

}

}

public string NoiSanXuat

{

get { return noiSanXuat; }

set

{

if (noiSanXuat.Length != 0)

{ noiSanXuat = value; }

}

}

public string DonViTinh

{

get { return donViTinh; }

set

{

if (donViTinh.Length != 0)

{ donViTinh = value; }

}

}

public string LoaiHang

{

get { return loaiHang; }

set

{

if (loaiHang.Length != 0)

{ loaiHang = value; }

}

}

public double GiaBan

{

get { return giaBan; }

set

{

if (giaBan >= 0)

{ giaBan = value; }

}

}

public DateTime HanSD

{

get { return hanSD; }

set

{

if (hanSD >= DateTime.Today)

{ hanSD = value; }

}

}

public virtual void Info()

{

Console.WriteLine($"Tên Hàng: {TenHang}.");

Console.WriteLine($"Mã Hàng: {MaHang}.");

Console.WriteLine($"Nơi Sản Xuất: {NoiSanXuat}.");

Console.WriteLine($"Đơn Vị Tính: {DonViTinh}.");

Console.WriteLine($"Loại Hàng: {LoaiHang}.");

Console.WriteLine($"Giá Bán: {GiaBan} VND.");

Console.WriteLine($"Hạn Sử Dụng: {HanSD.Day}/{HanSD.Month}/{HanSD.Year}");

}

}

}

Class HoaDon:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tieu\_Luan\_2.\_1.\_2

{

internal class HoaDon

{

private string diaChi, maHoaDon;

private string sDTCuaHang;

private double tongTien;

private DateTime ngayGio;

private NhanVien \_NhanVien;

private KhachHang \_KhachHang;

public HoaDon(string dc, string mhd, string sdtch, double tt, DateTime ng, NhanVien \_nv, KhachHang \_kh)

{

this.diaChi = dc; this.maHoaDon = mhd; this.sDTCuaHang = sdtch; this.tongTien = tt;

this.ngayGio = ng; this.\_NhanVien = \_nv; this.\_KhachHang = \_kh;

}

public string DiaChi => diaChi;

public string SDTCuaHang => sDTCuaHang;

public DateTime NgayGio => ngayGio;

public string MaHoaDon => maHoaDon;

public double TongTien

{

get { return tongTien; }

set { tongTien = value; }

}

public NhanVien NV => \_NhanVien;

public KhachHang KH => \_KhachHang;

public void Info()

{

Console.WriteLine("Thông tin hóa đơn: ");

Console.WriteLine($"Mã hóa đơn: {MaHoaDon}");

Console.WriteLine($"Mã Nhân viên: {NV.MaNV}");

Console.WriteLine($"Mã Khách Hàng : {KH.MaKH}");

Console.WriteLine($"Thời gian: {NgayGio.Day}/{NgayGio.Month}/{NgayGio.Year}");

Console.WriteLine($"Địa chỉ Siêu thị: {DiaChi}");

Console.WriteLine($"Tổng thanh toán: {TongTien}");

}

}

}

Class ChitietHoaDon:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tieu\_Luan\_2.\_1.\_2

{

internal class ChiTietHoaDon

{

private double soLuong, thanhTien;

private HangHoa hh;

private HoaDon hd;

public ChiTietHoaDon(double soLuong, double thanhTien, HangHoa hh, HoaDon hd)

{

this.soLuong = soLuong;

this.thanhTien = thanhTien;

this.hh = hh;

this.hd = hd;

}

public double SL => soLuong;

public double TT => thanhTien;

public HangHoa HH => hh;

public HoaDon HD => hd;

public string Center(string a, char b, int c)

{

int m = c - a.Length;

if (m <= 0) return a;

int left = (m / 2 + a.Length);

return a.PadLeft(left, b).PadRight(c, b);

}

public void Info()

{

Console.WriteLine($"||{Center(HH.MaHang, ' ', 20)}||{Center(HH.TenHang, ' ', 25)}||{Center(SL.ToString(), ' ', 10)}||{Center(HH.GiaBan.ToString(), ' ', 20)}||{Center(HH.DonViTinh, ' ', 20)}||{Center(TT.ToString(), ' ', 20)}||");

}

}

}

Class Controller:

using System;

using System.Collections.Generic;

using System.Globalization;

using System.Linq;

using System.Text.RegularExpressions;

using Tieu\_Luan\_2.\_1.\_2;

namespace HeThongQuanLyMuaBanSieuThi

{

internal class Controller

{

List<NhanVien> DSNV;

List<KhachHang> DSKH;

List<HoaDon> DSHD;

List<HangHoa> DSHH;

List<ChiTietHoaDon> DSCTHD;

List<CaLam> DSCL;

List<HoaDon> Da\_Nhap\_CT;

public Controller()

{

DSNV = new List<NhanVien>();

DSKH = new List<KhachHang>();

DSHD = new List<HoaDon>();

DSHH = new List<HangHoa>();

DSCTHD = new List<ChiTietHoaDon>();

DSCL = new List<CaLam>();

Da\_Nhap\_CT = new List<HoaDon>();

}

//CÁC LỆNH BỔ SUNG HỆ THỐNG

private string NhapDuLieuKhongKyTuDacBiet(string prompt)

{

if (string.IsNullOrWhiteSpace(prompt))

{

throw new ArgumentException("Dữ liệu không được bỏ trống!");

}

if (ContainsSpecialCharacters(prompt))

{

throw new ArgumentException("Dữ liệu không được chứa ký tự đặc biệt!");

}

return prompt;

}

private bool ContainsSpecialCharacters(string input)

{

return !Regex.IsMatch(input, @"^[a-zA-Z0-9\s]+$");

}

private bool IsValidDate(string date)

{

string[] dateParts = date.Split('/');

if (dateParts.Length != 3 ||

!int.TryParse(dateParts[0], out int day) ||

!int.TryParse(dateParts[1], out int month) ||

!int.TryParse(dateParts[2], out int year))

{

return false;

}

int maxDaysInMonth = DateTime.DaysInMonth(year, month);

return day >= 1 && day <= maxDaysInMonth &&

month >= 1 && month <= 12 &&

!(month == 2 && day > 29) &&

!(month == 2 && day == 29 && !DateTime.IsLeapYear(year));

}

public string Center(string a, char b, int c)

{

int m = c - a.Length;

if (m <= 0) return a;

int left = (m / 2 + a.Length);

return a.PadLeft(left, b).PadRight(c, b);

}

//LỆNH NHẬP VÀ THÊM

public NhanVien Nhap\_NV\_vathem()

{

Console.WriteLine("Nhập thông tin nhân viên");

Console.Write("Nhập Họ tên Nhân Viên: ");

string hoten\_x = Console.ReadLine();

if (string.IsNullOrEmpty(hoten\_x) || string.IsNullOrWhiteSpace(hoten\_x))

throw new Exception("Tên nhân viên không được bỏ trống!");

Console.Write("Nhập mã Nhân Viên: ");

string manv\_x = Console.ReadLine();

if (manv\_x.Length != 8) throw new Exception("Mã nhân viên phải có 8 kí tự!");

if (string.IsNullOrWhiteSpace(manv\_x)) throw new Exception("Mã nhân viên ko để trống!");

NhanVien nv = DSNV.FirstOrDefault(x => x.MaNV == manv\_x);

if (nv != null) throw new Exception("Mã Nhân Viên đã tồn tại trong danh sách");

Console.Write("Nhập Giới tính 0 - Nam || 1 - Nữ: ");

string gt = Console.ReadLine();

string gt\_x = "";

if (gt == "0") { gt\_x = "Nam"; }

else if (gt == "1") { gt\_x = "Nữ"; }

else { throw new Exception("Giới tính nhập không hợp lệ!"); }

Console.Write("Nhập số điện thoại nhân viên: ");

string phonenumber = Console.ReadLine();

if (!Int32.TryParse(phonenumber, out int y) || phonenumber.Length != 10 || !phonenumber.StartsWith("0")) throw new Exception("Số điện thoại không hợp lệ");

nv = DSNV.FirstOrDefault(x => x.SDT == phonenumber);

KhachHang kh = DSKH.FirstOrDefault(x => x.SDT == phonenumber);

if (nv != null || kh != null) throw new Exception("Số điện thoại đã tồn tại ");

Console.Write("Nhập địa chỉ của nhân viên: ");

string diachi\_x = Console.ReadLine();

if (string.IsNullOrWhiteSpace(diachi\_x)) throw new Exception("Địa chỉ không được để trống!");

Console.Write("Nhập ngày sinh nhân viên (Ngày/tháng/năm): ");

string ngaysinhInput = Console.ReadLine();

if (!IsValidDate(ngaysinhInput)) throw new Exception("Ngày sinh không hợp lệ!");

DateTime ngaysinh\_x = DateTime.ParseExact(ngaysinhInput, "dd/MM/yyyy", null);

if (DateTime.Now.Year - ngaysinh\_x.Year < 18) throw new Exception("Tuổi nhân viên chưa đủ 18!!");

Console.Write("Nhập chức vụ của nhân viên: ");

string chucvu\_x = Console.ReadLine();

if (string.IsNullOrWhiteSpace(chucvu\_x)) throw new Exception("Chức vụ chưa hợp lệ");

Console.Write("Nhập lương: ");

double luong\_x = Double.Parse(Console.ReadLine());

if (luong\_x < 0) throw new Exception("Lương phải lớn hơn 0 !");

Console.Write("Nhập phụ cấp: ");

string phucap\_x = Console.ReadLine();

NhanVien nv\_x = new NhanVien(hoten\_x, diachi\_x, gt\_x, phonenumber, ngaysinh\_x, manv\_x, chucvu\_x, phucap\_x, luong\_x);

DSNV.Add(nv\_x);

Console.WriteLine("Nhân viên đã được thêm vào danh sách!");

return nv\_x;

}

public KhachHang Nhap\_KH\_vathemKH()

{

Console.WriteLine("Nhập thông tin khách hàng");

Console.Write("Nhập Họ tên Khách Hàng: ");

string hoten\_x = Console.ReadLine();

if (string.IsNullOrEmpty(hoten\_x) || string.IsNullOrWhiteSpace(hoten\_x))

throw new Exception("Tên khách hàng không được bỏ trống!");

Console.Write("Nhập mã Khách Hàng: ");

string makh\_x = Console.ReadLine();

if (makh\_x.Length != 7) throw new Exception("Mã khách hàng phải có 7 kí tự!");

if (string.IsNullOrWhiteSpace(makh\_x)) throw new Exception("Mã khách hàng không được bỏ trống!");

KhachHang kh = DSKH.FirstOrDefault(x => x.MaKH == makh\_x);

if (kh != null) throw new Exception("mã KH đã tồn tại!");

Console.Write("Nhập Giới tính 0 - Nam || 1 - Nữ: ");

string gt = Console.ReadLine();

string gt\_x = "";

if (gt == "0") { gt\_x = "Nam"; }

else if (gt == "1") { gt\_x = "Nữ"; }

else { throw new Exception("Giới tính nhập không hợp lệ!"); }

Console.Write("Nhập số điện thoại khách hàng: ");

string phonenumber = Console.ReadLine();

if (!long.TryParse(phonenumber, out long y) || phonenumber.Length != 10 || !phonenumber.StartsWith("0")) throw new Exception("Số điện thoại không hợp lệ!");

kh = DSKH.FirstOrDefault(x => x.SDT == phonenumber); NhanVien nv = DSNV.FirstOrDefault(x => x.SDT == phonenumber);

if (nv != null || kh != null) throw new Exception("Số điện thoại đã tồn tại!");

Console.Write("Nhập địa chỉ của khách hàng: ");

string diachi\_x = Console.ReadLine();

Console.Write("Nhập ngày sinh khách hàng (Ngày/tháng/năm): ");

string ngaysinhInput = Console.ReadLine();

if (!IsValidDate(ngaysinhInput)) throw new Exception("Ngày sinh không hợp lệ!");

DateTime ngaysinh\_x = DateTime.ParseExact(ngaysinhInput, "dd/MM/yyyy", null);

if (DateTime.Now.Year - ngaysinh\_x.Year < 6) throw new Exception("Tuổi khách hàng chưa đủ 6 tuổi!");

Console.Write("Nhập điểm tích lũy của khách hàng: ");

if (!Int32.TryParse(Console.ReadLine(), out int diem\_tich\_luy\_x) || diem\_tich\_luy\_x < 0 || diem\_tich\_luy\_x > 1000)

throw new Exception("Điểm tích lũy không hợp lệ!");

KhachHang kh\_x = new KhachHang(hoten\_x, diachi\_x, gt\_x, phonenumber, ngaysinh\_x, makh\_x, diem\_tich\_luy\_x);

DSKH.Add(kh\_x);

Console.WriteLine($"Khách hàng có mã {makh\_x} đã được thêm vào danh sách");

return kh\_x;

}

public HoaDon Nhap\_HoaDon\_vathem()

{

Console.WriteLine("Nhập thông tin hóa đơn");

Console.Write("Nhập mã hóa đơn: ");

string maHD = Console.ReadLine();

if (string.IsNullOrWhiteSpace(maHD)) throw new Exception("Mã hóa đơn không được bỏ trống!");

HoaDon t\_1=DSHD.FirstOrDefault(x=>x.MaHoaDon == maHD);

if (t\_1 != null) throw new Exception("mã hóa đơn đã tồn tại");

Console.Write("Nhập ngày lập hóa đơn (Ngày/tháng/năm): ");

string ngayGio = Console.ReadLine();

DateTime ngayGioInput = DateTime.Parse(ngayGio);

if (!IsValidDate(ngayGio)) throw new Exception("Ngày lập hóa đơn không hợp lệ!");

DateTime ngayLapHD = DateTime.ParseExact(ngayGio, "dd/MM/yyyy", null);

Console.Write("Nhập mã khách hàng: ");

string maKH = Console.ReadLine();

if (string.IsNullOrWhiteSpace(maKH)) throw new Exception("Mã khách hàng không được bỏ trống!");

KhachHang khachHang = DSKH.FirstOrDefault(kh => kh.MaKH == maKH);

if (khachHang == null) throw new Exception("Mã Khách Hàng chưa tồn tại ");

Console.Write("Nhập số điện thoại cửa hàng: ");

string sDTCuaHang = Console.ReadLine();

long sDTCuaHangInput = long.Parse(sDTCuaHang);

if (string.IsNullOrWhiteSpace(sDTCuaHang)) throw new Exception("Số điện thoại cửa hàng không được bỏ trống!");

if (sDTCuaHang.Length != 10 || !long.TryParse(sDTCuaHang, out sDTCuaHangInput)) throw new Exception("Số điện thoại chưa hợp lệ");

NhanVien n\_v = DSNV.FirstOrDefault(x => x.SDT == sDTCuaHang);

KhachHang k\_h = DSKH.FirstOrDefault(x => x.SDT == sDTCuaHang);

if (n\_v != null || k\_h != null) throw new Exception("Số điện thoại đã tồn tại");

Console.Write("Nhập mã nhân viên lập hóa đơn: ");

string maNV = Console.ReadLine();

if (string.IsNullOrWhiteSpace(maNV)) throw new Exception("Mã nhân viên không được bỏ trống!");

NhanVien nhanVien = DSNV.FirstOrDefault(nv => nv.MaNV == maNV);

if (nhanVien == null) throw new Exception("Không tìm thấy nhân viên với mã này!");

Console.Write("Nhập địa chỉ Siêu Thị: ");

string diaChi = Console.ReadLine();

if (string.IsNullOrWhiteSpace(diaChi)) throw new Exception("Địa chỉ Siêu thị không được bỏ trống!");

Console.Write("Nhập tổng tiền hóa đơn: ");

string tongTienInput = Console.ReadLine();

if (string.IsNullOrWhiteSpace(tongTienInput)) throw new Exception("Tổng tiền hóa đơn không được bỏ trống!");

double tongTien = double.Parse(tongTienInput);

if (tongTien <= 0) throw new Exception("Tổng tiền hóa đơn phải là một số dương!");

HoaDon hoaDon = new HoaDon(diaChi, maHD, sDTCuaHang, tongTien, ngayGioInput, nhanVien, khachHang);

DSHD.Add(hoaDon);

Console.WriteLine("Thêm hóa đơn thành công!");

return hoaDon;

}

public HangHoa Nhap\_HangHoa\_vathem()

{

Console.WriteLine("Nhập thông tin hàng hóa");

Console.Write("Nhập tên hàng: ");

string tenHang = Console.ReadLine();

if (string.IsNullOrWhiteSpace(tenHang)) throw new Exception("Tên hàng không được bỏ trống!");

Console.Write("Nhập mã hàng hóa: ");

string maHang = Console.ReadLine();

if (string.IsNullOrWhiteSpace(maHang)) throw new Exception("Mã hàng hóa không được bỏ trống!");

Console.Write("Nhập nơi sản xuất: ");

string noiSanXuat = Console.ReadLine();

if (string.IsNullOrWhiteSpace(noiSanXuat))

throw new ArgumentException("Nơi sản xuất không được bỏ trống!");

Console.Write("Nhập đơn vị tính: ");

string donViTinh = Console.ReadLine();

if (string.IsNullOrWhiteSpace(donViTinh))

throw new ArgumentException("Đơn vị tính không được bỏ trống!");

Console.Write("Nhập loại hàng: ");

string loaiHang = Console.ReadLine();

Console.Write("Nhập giá bán: ");

double giaBan;

if (!double.TryParse(Console.ReadLine(), out giaBan) || giaBan <= 0)

throw new ArgumentException("Giá bán phải là một số dương!");

Console.Write("Nhập hạn sử dụng (Ngày/tháng/năm): ");

string[] hanSuDung = Console.ReadLine().Split('/');

DateTime HSD = new DateTime(Int32.Parse(hanSuDung[2]), Int32.Parse(hanSuDung[1]), Int32.Parse(hanSuDung[0]));

HangHoa hangHoa = new HangHoa(tenHang, maHang, noiSanXuat, donViTinh, loaiHang, giaBan, HSD);

DSHH.Add(hangHoa);

Console.WriteLine("Thêm hàng hóa thành công!");

return hangHoa;

}

public CaLam Nhap\_CL\_vatthem()

{

Console.WriteLine("Nhập thông tin ca làm");

Console.Write("Nhập ID Ca: ");

string idCa = Console.ReadLine();

if (string.IsNullOrWhiteSpace(idCa))

throw new ArgumentException("ID Ca không được bỏ trống!");

if (idCa.Length != 6)

throw new Exception("ID ca phải có 6 kí tự!");

Console.Write("Nhập Tên Ca: ");

string tenCa = Console.ReadLine();

if (string.IsNullOrWhiteSpace(tenCa))

throw new ArgumentException("Tên Ca không được bỏ trống!");

Console.Write("Nhập Ngày làm (Ngày/tháng/năm): ");

DateTime ngayLam;

string input = Console.ReadLine();

if (!DateTime.TryParseExact(input, "dd/MM/yyyy", null, DateTimeStyles.None, out ngayLam))

throw new ArgumentException("Ngày làm không hợp lệ!");

if (ngayLam.Date < DateTime.Now.Date)

throw new ArgumentException("Ngày làm không được nhỏ hơn ngày hiện tại!");

Console.Write("Nhập Giờ làm (HH:mm): ");

DateTime gioLam;

string gioLamInput = Console.ReadLine();

if (!DateTime.TryParseExact(gioLamInput, "HH:mm", CultureInfo.InvariantCulture, DateTimeStyles.None, out gioLam))

throw new Exception("Giờ làm không hợp lệ!");

if (gioLam.TimeOfDay.TotalHours < 0 || gioLam.TimeOfDay.TotalHours > 24)

throw new ArgumentException("Giờ làm không hợp lệ!");

Console.Write("Nhập Giờ nghỉ giữa ca (HH:mm): ");

DateTime gioNghiGiuaCa;

string gioNghiGiuaCaInput = Console.ReadLine();

if (!DateTime.TryParseExact(gioNghiGiuaCaInput, "HH:mm", CultureInfo.InvariantCulture, DateTimeStyles.None, out gioNghiGiuaCa))

throw new Exception("Giờ nghỉ giữa ca không hợp lệ!");

if (gioNghiGiuaCa.TimeOfDay.TotalHours < 0 || gioNghiGiuaCa.TimeOfDay.TotalHours > 24)

throw new Exception("Giờ nghỉ giữa ca không hợp lệ!");

Console.Write("Nhập Mã Nhân viên: ");

string maNV = Console.ReadLine();

if (string.IsNullOrWhiteSpace(maNV)) throw new Exception("Mã nhân viên không được bỏ trống!");

NhanVien nhanVien = DSNV.FirstOrDefault(nv => nv.MaNV == maNV);

if (maNV.Length != 8) throw new Exception("Mã nhân viên phảo có 8 kí tự!");

CaLam caLam = new CaLam(nhanVien, idCa, tenCa, ngayLam, gioLam, gioNghiGiuaCa);

DSCL.Add(caLam);

Console.WriteLine("Thêm ca làm thành công!");

return caLam;

}

public void Nhap\_CTHD\_cho\_MaHD\_naodo()

{

Console.Write("Nhập mã Hóa đơn Tương ứng: ");

string k = Console.ReadLine();

HoaDon hd = DSHD.FirstOrDefault(x => x.MaHoaDon == k);

if (hd == null) throw new Exception("Mã Hóa đơn chưa có trong danh sách!");

HoaDon hdDaNhap = Da\_Nhap\_CT.FirstOrDefault(x => x.MaHoaDon == k);

if (hdDaNhap != null) throw new Exception("Hóa đơn này đã được thêm chi tiết ");

double TT\_new = hd.TongTien;

HangHoa h = null;

double sl = 0;

double total = 0;

try

{

while (true)

{

Console.Write("Nhập mã Hàng Hóa (nhấn 'exit' nếu đã hoàn thành việc nhập): ");

string ma = Console.ReadLine();

h = DSHH.FirstOrDefault(x => x.MaHang == ma);

if (h != null)

{

Console.WriteLine($"Tên hàng: {h.TenHang} Đơn vị: {h.DonViTinh}");

Console.Write("Nhập số lượng hàng: ");

if (!Double.TryParse(Console.ReadLine(), out sl) || sl <= 0) throw new Exception("Số lượng là 1 số thực lớn hơn không!");

ChiTietHoaDon ct1 = new ChiTietHoaDon(sl, sl \* h.GiaBan, h, hd);

DSCTHD.Add(ct1);

total += sl \* h.GiaBan;

}

else if (ma == "exit")

{

if (total != TT\_new)

{

throw new Exception("Tổng tiền chưa hợp lệ ");

}

else

{

Da\_Nhap\_CT.Add(hd);

Console.WriteLine("Thêm chi tiết vào HD thành công");

break;

}

}

else

{

throw new Exception("Mã hàng hóa không tồn tại ! ");

}

}

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

}

public void HienThi\_NV\_Theo\_MaNV(string z)

{

bool found = false;

foreach (var nv in DSNV)

{

if (nv.MaNV == z)

{

nv.Info();

found = true;

break;

}

}

if (!found) throw new Exception("Nhân viên không tồn tại trong danh sách!");

}

public void HienThi\_KH\_Theo\_MaKH(string z)

{

bool found = false;

foreach (var kh in DSKH)

{

if (kh.MaKH == z)

{

kh.Info();

found = true;

break;

}

}

if (!found) throw new Exception("Khách hàng không tồn tại trong danh sách!");

}

public void HienThi\_HD\_Theo\_MaHD(string z)

{

bool found = false;

foreach (var hd in DSHD)

{

if (hd.MaHoaDon == z)

{

hd.Info();

found = true;

break;

}

}

if (!found) throw new Exception("Hóa đơn không tồn tại trong danh sách!");

}

public void HienThi\_HH\_Theo\_MaHH(string z)

{

bool found = false;

foreach (var hh in DSHH)

{

if (hh.MaHang == z)

{

hh.Info();

found = true;

break;

}

}

if (!found) throw new Exception("Hóa đơn không tồn tại trong danh sách!");

}

public void Hien\_Thi\_Ca\_Lam\_Bang\_ID\_Ca(string z)

{

foreach (CaLam i in DSCL)

{

if (i.IDCa == z)

{

i.Info();

}

else

throw new Exception ("Không có Ca Làm hợp lệ với ID bạn vừa nhập!");

}

}

public void IN\_CTHD\_Theo\_maHD(string z)

{

HoaDon hd = Da\_Nhap\_CT.FirstOrDefault(x => x.MaHoaDon == z);

if (hd == null) throw new Exception("Mã hóa đơn chưa được nhập chi tiết");

Console.WriteLine("");

Console.WriteLine($"||{Center("===", '=', 125)}||");

Console.WriteLine($"||{Center(" ", ' ', 125)}||");

Console.WriteLine($"||{Center("= Chi Tiết Hóa Đơn =", ' ', 125)}||");

Console.WriteLine($"||{Center("Địa chỉ: " + hd.DiaChi, ' ', 125)}||");

Console.WriteLine($"||{Center("Mã Hóa đơn: " + hd.MaHoaDon, ' ', 125)}||");

Console.WriteLine($"||{Center("Time: " + hd.NgayGio.ToString("dd/MM/yyyy"), ' ', 125)}||");

Console.WriteLine($"||{Center("---", '-', 125)}||");

Console.WriteLine($"||{Center("Mã Hàng", ' ', 20)}||{Center("Tên Hàng", ' ', 25)}||{Center("Số Lượng", ' ', 10)}||{Center("Giá Bán", ' ', 20)}||{Center("Đơn vị tính", ' ', 20)}||{Center("Thành Tiền", ' ', 20)}||");

Console.WriteLine($"||{Center("---", '-', 125)}||");

foreach (var i in DSCTHD)

{

if (i.HD.MaHoaDon == z)

{

i.Info();

Console.WriteLine($"||{Center("---", '-', 125)}||");

}

}

Console.WriteLine($"||{Center("TỔNG CỘNG - TOTAL ", ' ', 102)}||{Center(hd.TongTien.ToString(), ' ', 20)}||");

Console.WriteLine($"||{Center("===", '=', 125)}||");

}

public void Check\_Xoa\_CTHD\_MaHD(string z)

{

foreach (var i in DSCTHD)

{

if (i.HD.MaHoaDon == z)

{

DSCTHD.Remove(i);

}

}

}

public void Check\_Xoa\_HD\_MaHD(string k)

{

HoaDon hd = DSHD.FirstOrDefault(x => x.MaHoaDon == k);

if (hd != null)

{

DSHD.Remove(hd);

Console.WriteLine($"Hóa đơn Mã: {hd.MaHoaDon} đã bị xóa khỏi danh sách!");

Check\_Xoa\_CTHD\_MaHD(k);

}

else throw new Exception("Mã hóa đơn không tồn tại!");

}

public void Xoa\_NV\_MaNV()

{

Console.Write("Nhập mã nhân viên: ");

string a = Console.ReadLine();

NhanVien nv = DSNV.FirstOrDefault(x => x.MaNV == a);

if (nv != null)

{

Console.WriteLine("Xóa nhân viên thành công!");

DSNV.Remove(nv);

}

else

{

throw new Exception("Không tìm thấy nhân viên để xóa!");

}

}

public void Xoa\_Calam\_ID\_K(string id)

{

CaLam caLam = DSCL.FirstOrDefault(clo => clo.IDCa == id);

if (caLam != null)

{

DSCL.Remove(caLam);

Console.WriteLine($"Ca làm có ID: {caLam.IDCa} đã xóa khỏi danh sách!");

}

else throw new Exception("ID Ca Làm không tồn tại!");

}

public void Xoa\_KH\_MaKH(string maKH)

{

KhachHang kh = DSKH.FirstOrDefault(x => x.MaKH == maKH);

if (kh != null)

{

DSKH.Remove(kh);

Console.WriteLine("Xóa khách hàng thành công!");

}

else

{

Console.WriteLine("Không tìm thấy khách hàng để xóa!");

}

}

public void Xoa\_HH\_MaHang(string MaHH)

{

HangHoa hh = DSHH.FirstOrDefault(x => x.MaHang == MaHH);

if (hh != null)

{

DSHH.Remove(hh);

Console.WriteLine($"Hàng hóa có mã {hh.MaHang} đã bị xóa khỏi DS Hàng Hóa ");

}

else throw new Exception("Không tìm thấy hàng hóa");

}

public void In\_NV()

{

int j = 1;

Console.WriteLine();

Console.WriteLine($"{Center("Danh Sách Nhân Viên", ' ', 85)}");

Console.WriteLine($"{Center("= =", '=', 85)}");

Console.WriteLine($"||{Center("STT", ' ', 5)}||{Center("Tên Nhân viên", ' ', 30)}||{Center("Mã Nhân viên", ' ', 20)}||{Center("Chức vụ", ' ', 20)}||");

Console.WriteLine($"{Center("---", '-', 85)}");

foreach (var i in DSNV)

{

Console.WriteLine($"||{Center(j.ToString(), ' ', 5)}||{Center(i.HoTen, ' ', 30)}||{Center(i.MaNV, ' ', 20)}||{Center(i.ChucVu, ' ', 20)}||"); j += 1;

}

Console.WriteLine($"{Center("---", '-', 85)}");

}

public void In\_HH()

{

int j = 1;

Console.WriteLine();

Console.WriteLine($"{Center("Danh Sách Hàng Hóa", ' ', 77)}");

Console.WriteLine($"{Center("= =", '=', 77)}");

Console.WriteLine($"||{Center("STT", ' ', 5)}||{Center("Tên Hàng", ' ', 20)}||{Center("Mã Hàng", ' ', 13)}||{Center("Giá bán", ' ', 15)}||{Center("Đơn vị tính", ' ', 13)}||");

Console.WriteLine($"{Center("---", '-', 77)}");

foreach (var i in DSHH)

{

Console.WriteLine($"||{Center(j.ToString(), ' ', 5)}||{Center(i.TenHang, ' ', 20)}||{Center(i.MaHang, ' ', 13)}||{Center(i.GiaBan.ToString(), ' ', 15)}||{Center(i.DonViTinh, ' ', 13)}||"); j += 1;

}

Console.WriteLine($"{Center("---", '-', 77)}");

}

public void In\_KH()

{

int j = 1;

Console.WriteLine();

Console.WriteLine($"{Center("Danh Sách Khách Hàng", ' ', 85)}");

Console.WriteLine($"{Center("= =", '=', 85)}");

Console.WriteLine($"||{Center("STT", ' ', 5)}||{Center("Tên Khách Hàng", ' ', 30)}||{Center("Mã Khách hàng", ' ', 20)}||{Center("Điểm tích lũy", ' ', 20)}||");

Console.WriteLine($"{Center("---", '-', 85)}");

foreach (var i in DSKH)

{

Console.WriteLine($"||{Center(j.ToString(), ' ', 5)}||{Center(i.HoTen, ' ', 30)}||{Center(i.MaKH, ' ', 20)}||{Center(i.DiemTichLuy.ToString(), ' ', 20)}||"); j += 1;

}

Console.WriteLine($"{Center("---", '-', 85)}");

}

public void In\_CL()

{

int j = 1;

Console.WriteLine();

Console.WriteLine($"{Center("Danh Sách Ca Làm", ' ', 43)}");

Console.WriteLine($"{Center("= =", '=', 43)}");

Console.WriteLine($"||{Center("STT", ' ', 5)}||{Center("ID Ca", ' ', 10)}||{Center("Thời Gian", ' ', 20)}||");

Console.WriteLine($"{Center("---", '-', 43)}");

foreach (var i in DSCL)

{

Console.WriteLine($"||{Center(j.ToString(), ' ', 5)}||{Center(i.IDCa, ' ', 10)}||{Center(i.NgayLam.ToString("dd/MM/yyyy")+" "+i.GioLam.ToString("HH:mm"), ' ', 20)}||");

j += 1;

}

Console.WriteLine($"{Center("---", '-', 43)}");

}

public void In\_Ma\_HD()

{

int j = 1;

foreach (var i in DSHD)

{

Console.WriteLine($"{j}. Mã hóa đơn: {i.MaHoaDon}");

j += 1;

}

}

public void CapNhat\_KH(string MaKH)

{

KhachHang kh = DSKH.FirstOrDefault(x => x.MaKH == MaKH);

if (kh != null)

{

kh.Info();

DSKH.Remove(kh);

Console.WriteLine("Khách hàng sẽ bị xóa, hãy cập nhật thông tin khách hàng này");

int a = DSKH.IndexOf(kh);

Nhap\_KH\_vathemKH();

}

else throw new Exception("Mã khách hang không tồn tại");

}

public void CapNhat\_HH(string MaHang)

{

HangHoa hh = DSHH.FirstOrDefault(x => x.MaHang == MaHang);

if (hh != null)

{

hh.Info();

DSHH.Remove(hh);

Console.WriteLine("Hàng hóa sẽ bị xóa, hãy cập nhật thông tin Hàng hóa này"); Nhap\_HangHoa\_vathem();

}

else throw new Exception("Mã hàng hóa không tồn tại");

}

public void CapNhat\_NV(string MaNV)

{

NhanVien nv = DSNV.FirstOrDefault(x => x.MaNV == MaNV);

if (nv != null)

{

nv.Info();

DSNV.Remove(nv);

Console.WriteLine("Nhân Viên này sẽ bị xóa, hãy cập nhật thông tin Nhân viên này");

Nhap\_NV\_vathem();

}

else throw new Exception("Mã Nhân Viên không tồn tại");

}

public void CapNhat\_CL(string z)

{

CaLam clo = DSCL.FirstOrDefault(x => x.IDCa == z);

if (clo != null)

{

clo.Info();

DSCL.Remove(clo);

Console.WriteLine("Ca Làm này sẽ bị xóa, hãy cập nhật thông tin Ca Làm này");

Nhap\_CL\_vatthem();

}

else throw new Exception("ID Ca Làm không tồn tại");

}

public void Menu\_Va\_Chay\_CT()

{

Controller ctr = new Controller();

Console.WriteLine("|===================================================================|");

Console.WriteLine("| HỆ THỐNG QUẢN LÝ |");

Console.WriteLine("| MUA BÁN SIÊU THỊ |");

Console.WriteLine("|===================================================================|");

Console.WriteLine("| MENU CHỨC NĂNG |");

Console.WriteLine("| |");

Console.WriteLine("| AA: Nhập thông tin và thêm: BB: Hiển Thị Thông tin|");

Console.WriteLine("| 01. Nhân viên mới 01. Nhân Viên |");

Console.WriteLine("| 02. Khách hàng mới 02. Khách hàng |");

Console.WriteLine("| 03. Hóa đơn mới 03. Hóa đơn |");

Console.WriteLine("| 04. Hàng hóa mới 04. Hàng hóa |");

Console.WriteLine("| 05. Ca làm mới 05. Ca làm |");

Console.WriteLine("| 06. Chi tiết HD theo Mã HD 06. Chi tiết HD |");

Console.WriteLine("| |");

Console.WriteLine("| CC:Xóa thông tin |");

Console.WriteLine("| [ Các dữ liệu liên quan DD:Cập nhật |");

Console.WriteLine("| có thể sẽ không thể 01: Nhân Viên |");

Console.WriteLine("| truy cập, hãy cân nhắc ] 02: Khách Hàng |");

Console.WriteLine("| 01. Nhân Viên 03: Hàng Hóa |");

Console.WriteLine("| 02. Khách Hàng 04: Ca Làm |");

Console.WriteLine("| 03. Ca Làm 00:Thoát Chương trình |");

Console.WriteLine("| 04. Hóa đơn |");

Console.WriteLine("| 05. Hàng Hóa |");

Console.WriteLine("| |");

Console.WriteLine("| EE: IN DANH SÁCH |");

Console.WriteLine("| 01: Mã Nhân Viên |");

Console.WriteLine("| 02: Mã Khách Hàng |");

Console.WriteLine("| 03: Mã Hàng Hóa |");

Console.WriteLine("| 04: ID Ca Làm |");

Console.WriteLine("| 05: Mã Hóa đơn |");

Console.WriteLine("| |");

Console.WriteLine("|Lưu ý: Cứ pháp để thực hiện: |");

Console.WriteLine("| <Tên Mục>-<số có lệnh bạn muốn thực hiện> |");

Console.WriteLine("| VD: AA-01 <Thêm Nhân viên mới> |");

Console.WriteLine("|===================================================================|");

ctr.DS\_Tam\_Thoi();

while (true)

{

Console.Write("Nhập lựa chọn của bạn dựa trên MENU chức năng: ");

string result = Console.ReadLine();

int sogach = result.Count(x => x == '-');

if (sogach == 1)

{

string[] choice = result.Split('-');

string k = choice[0];

string v = choice[1];

if (choice.Length == 2)

{

foreach (string z in choice)

{

try

{

ctr.NhapDuLieuKhongKyTuDacBiet(z);

ctr.ContainsSpecialCharacters(z);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

Console.ReadKey();

}

}

switch (k)

{

case "AA":

if (v == "01")

{

try

{

ctr.Nhap\_NV\_vathem();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

Console.ReadKey();

}

}

else if (v == "02")

{

try

{

ctr.Nhap\_KH\_vathemKH();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

Console.ReadKey();

}

}

else if (v == "03")

{

try

{

ctr.Nhap\_HoaDon\_vathem();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

Console.ReadKey();

}

}

else if (v == "04")

{

try

{

ctr.Nhap\_HangHoa\_vathem();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

Console.ReadKey();

}

}

else if (v == "05")

{

try

{

ctr.Nhap\_CL\_vatthem();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

Console.ReadKey();

}

}

else if (v == "06")

{

try

{

ctr.Nhap\_CTHD\_cho\_MaHD\_naodo();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

Console.ReadKey();

}

}

else

{

try { throw new Exception("Lỗi cú pháp"); }

catch (Exception y)

{

Console.WriteLine(y.Message);

Console.ReadKey();

}

}

break;

case "BB":

if (v == "01")

{

try

{

Console.Write("Nhập mã nhân viên bạn muốn in ra thông tin: ");

ctr.HienThi\_NV\_Theo\_MaNV(Console.ReadLine());

}

catch (Exception e)

{

Console.WriteLine(e.Message);

Console.ReadKey();

}

}

else if (v == "02")

{

try

{

Console.Write("Nhập mã Khách Hàng bạn muốn in ra thông tin: ");

ctr.HienThi\_KH\_Theo\_MaKH(Console.ReadLine());

}

catch (Exception e)

{

Console.WriteLine(e.Message);

Console.ReadKey();

}

}

else if (v == "03")

{

try

{

Console.Write("Nhập mã Hóa Đơn bạn muốn in thông tin: ");

ctr.HienThi\_HD\_Theo\_MaHD(Console.ReadLine());

}

catch (Exception e)

{

Console.WriteLine(e.Message);

Console.ReadKey();

}

}

else if (v == "04")

{

try

{

Console.Write("NHập mã Hàng hóa bạn muốn in thông tin: ");

ctr.HienThi\_HH\_Theo\_MaHH(Console.ReadLine());

}

catch (Exception e)

{

Console.WriteLine(e.Message);

Console.ReadKey();

}

}

else if (v == "05")

{

try

{

Console.Write("Nhập ID Ca Làm bạn muốn in thông tin: ");

ctr.Hien\_Thi\_Ca\_Lam\_Bang\_ID\_Ca(Console.ReadLine());

}

catch (Exception e)

{

Console.WriteLine(e.Message);

Console.ReadKey();

}

}

else if (v == "06")

{

try

{

Console.Write("NHập mã hóa đơn bạn muốn in ra thông tin chi tiết hóa đơn: ");

ctr.IN\_CTHD\_Theo\_maHD(Console.ReadLine());

}

catch (Exception e)

{

Console.WriteLine(e.Message);

Console.ReadKey();

}

}

else

{

try { throw new Exception("Lỗi cú pháp"); }

catch (Exception y)

{

Console.WriteLine(y.Message);

Console.ReadKey();

}

}

break;

case "CC":

if (v == "01")

{

try

{

ctr.Xoa\_NV\_MaNV();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message); Console.ReadKey();

}

}

else if (v == "02")

{

try

{

Console.Write("NHập mã KH cần xóa: ");

ctr.Xoa\_KH\_MaKH(Console.ReadLine());

}

catch (Exception ex)

{

Console.WriteLine(ex.Message); Console.ReadKey();

}

}

else if (v == "03")

{

try

{

Console.Write("NHập ID Ca Làm cần xóa: ");

ctr.Xoa\_Calam\_ID\_K(Console.ReadLine());

}

catch (Exception ex)

{

Console.WriteLine(ex.Message); Console.ReadKey();

}

}

else if (v == "04")

{

try

{

Console.Write("NHập mã HD cần xóa: ");

ctr.Check\_Xoa\_HD\_MaHD(Console.ReadLine());

}

catch (Exception ex)

{

Console.WriteLine(ex.Message); Console.ReadKey();

}

}

else if (v == "05")

{

try

{

Console.Write("NHập mã Hàng Hóa cần xóa: ");

ctr.Xoa\_HH\_MaHang(Console.ReadLine());

}

catch (Exception ex)

{

Console.WriteLine(ex.Message); Console.ReadKey();

}

}

else

{

try { throw new Exception("Lỗi cú pháp"); }

catch (Exception y)

{

Console.WriteLine(y.Message);

Console.ReadKey();

}

}

break;

case "DD":

if (v == "01")

{

Console.WriteLine("Lưu ý: Cập nhật thì hãy dựa trên thông tin cũ, cái nào thay đổi thì sửa, còn không thì cũng nhập lại giống cái cũ");

try

{

Console.Write("NHập Mã Nhân viên cần cập nhật: ");

ctr.CapNhat\_NV(Console.ReadLine());

}

catch (Exception ex)

{

Console.WriteLine(ex.Message); Console.ReadKey();

}

}

else if (v == "02")

{

Console.WriteLine("Lưu ý: Cập nhật thì hãy dựa trên thông tin cũ, cái nào thay đổi thì sửa, còn không thì cũng nhập lại giống cái cũ");

try

{

Console.Write("NHập Mã Khách Hàng cần cập nhật: ");

ctr.CapNhat\_KH(Console.ReadLine());

}

catch (Exception ex)

{

Console.WriteLine(ex.Message); Console.ReadKey();

}

}

else if (v == "03")

{

Console.WriteLine("Lưu ý: Cập nhật thì hãy dựa trên thông tin cũ, cái nào thay đổi thì sửa, còn không thì cũng nhập lại giống cái cũ");

try

{

Console.Write("NHập Mã Hàng Hóa cần cập nhật: ");

ctr.CapNhat\_HH(Console.ReadLine());

}

catch (Exception ex)

{

Console.WriteLine(ex.Message); Console.ReadKey();

}

}

else if (v == "04")

{

Console.WriteLine("Lưu ý: Cập nhật thì hãy dựa trên thông tin cũ, cái nào thay đổi thì sửa, còn không thì cũng nhập lại giống cái cũ");

try

{

Console.Write("NHập ID Ca làm cần cập nhật: ");

ctr.CapNhat\_CL(Console.ReadLine());

}

catch (Exception ex)

{

Console.WriteLine(ex.Message); Console.ReadKey();

}

}

else if (v == "00")

{

Console.WriteLine("chương trình sẽ kết thức và dữ liệu sẽ không dược sử dụng ở chương trình mới");

Console.ReadKey();

Console.Clear();

}

else

{

try { throw new Exception("Lỗi cú pháp"); }

catch (Exception y)

{

Console.WriteLine(y.Message);

Console.ReadKey();

}

}

break;

case "EE":

if (v == "01")

{

try

{

ctr.In\_NV();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

Console.ReadLine();

}

}

else if (v == "02")

{

try

{

ctr.In\_KH();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

Console.ReadLine();

}

}

else if (v == "03")

{

try

{

ctr.In\_HH();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

Console.ReadLine();

}

}

else if (v == "04")

{

try

{

ctr.In\_CL();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

Console.ReadLine();

}

}

else if (v == "05")

{

try

{

ctr.In\_Ma\_HD();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

Console.ReadLine();

}

}

else

{

try { throw new Exception("Lỗi cú pháp"); }

catch (Exception y)

{

Console.WriteLine(y.Message);

Console.ReadKey();

}

}

break;

default:

{

try { throw new Exception("Lỗi cú pháp"); }

catch (Exception y)

{

Console.WriteLine(y.Message);

Console.ReadKey();

}

}

break;

}

}

}

else

{

try { throw new Exception("Lỗi cú pháp"); }

catch (Exception y)

{

Console.WriteLine(y.Message);

Console.ReadKey();

}

}

}

}

public void DS\_Tam\_Thoi()

{

NhanVien nv = new NhanVien("Nguyễn Tuấn ANh", "Thuận AN", "Nam", "0328884320", new DateTime(2005, 10, 06), "23050118", "Thu Ngân", "Không", 500000);

KhachHang kh = new KhachHang("Nguyễn Tuấn ANh", "Thuận AN", "Nam", "0328884322", new DateTime(2005, 10, 06), "2305011", 150);

HangHoa pork = new HangHoa("Thịt lợn", "001", "2023", "Ký", "Nạc", 85000, new DateTime(2024, 12, 12));

HangHoa fish = new HangHoa("Cá", "002", "2023", "Ký", "Ba Sa", 120000, new DateTime(2024, 12, 12));

HangHoa cuahoangde = new HangHoa("Cua vjp", "003", "2023", "Ký", "Hoàng đế", 3000000, new DateTime(2024, 12, 12));

HoaDon hd = new HoaDon("New York", "123", "0123456789", 3205000, DateTime.Now, nv, kh);

DSNV.Add(nv);

DSKH.Add(kh);

DSHH.Add(pork); DSHH.Add(fish); DSHH.Add(cuahoangde);

DSHD.Add(hd);

}

}

}

Class Program:

using HeThongQuanLyMuaBanSieuThi;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tieu\_Luan\_2.\_1.\_2

{

internal class Program

{

static void Main(string[] args)

{

Console.InputEncoding = Encoding.UTF8;

Console.OutputEncoding = Encoding.UTF8;

Controller controller = new Controller();

controller.Menu\_Va\_Chay\_CT();

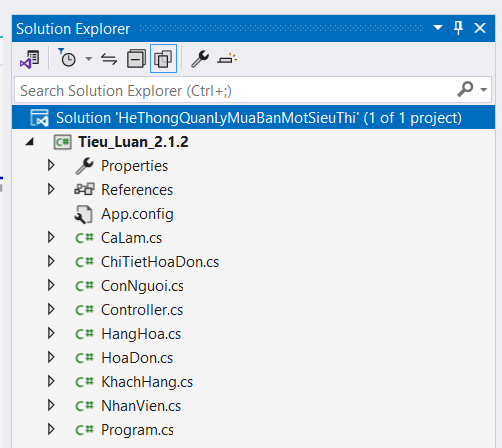
}

}

}

## Thử nghiệm (Testing)

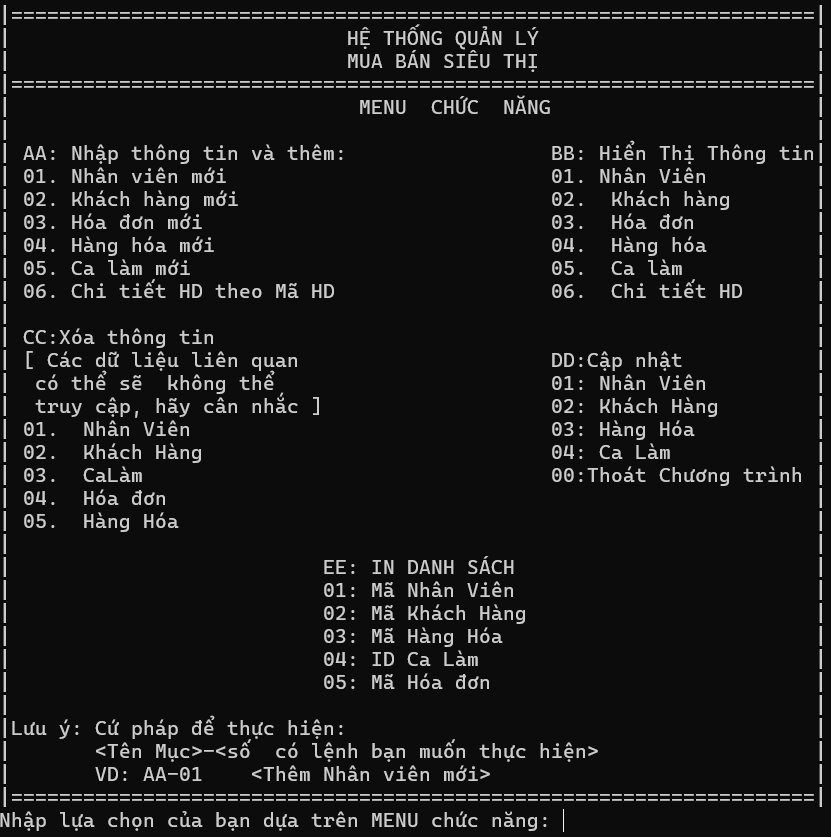
Ảnh chụp màn hình MS Visual Studio có thể hiện Solution /Project của chương trình



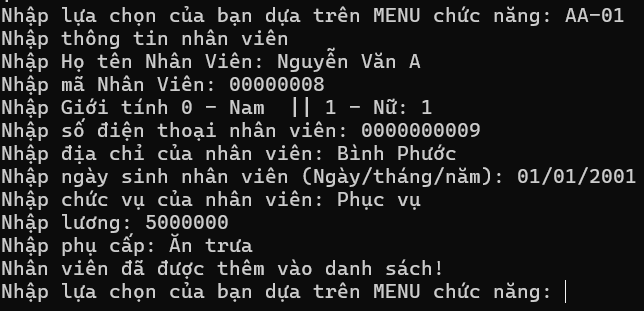
Ảnh chụp màn hình kết quả khởi động chương trình, kết quả chạy từng chức năng của

chương trình:

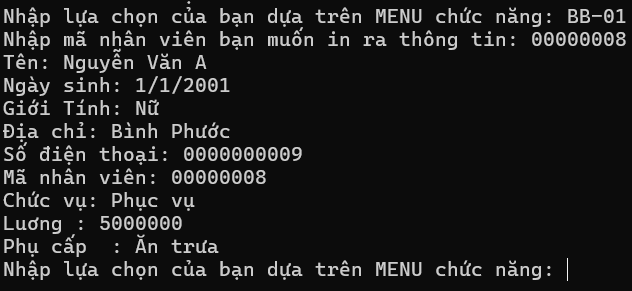
Các chức năng menu chính của chương trình:



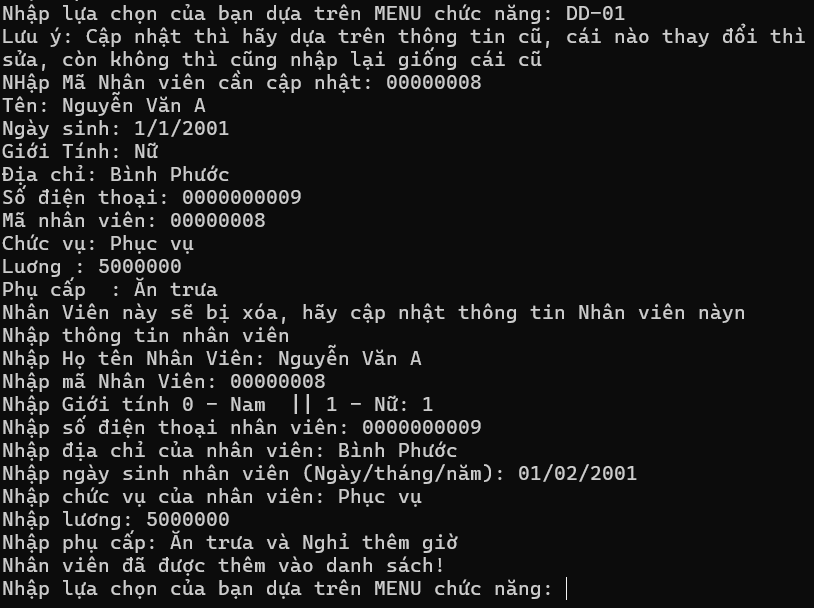
AA-01:Nhập thông tin và thêm Nhân viên mới



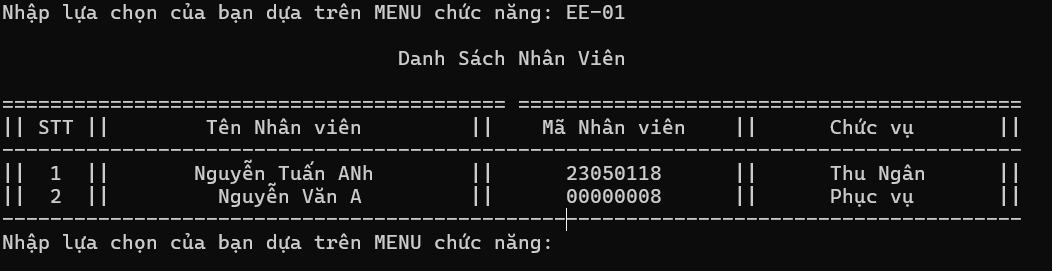
BB-01:Hiển thị thông tin Nhân Viên



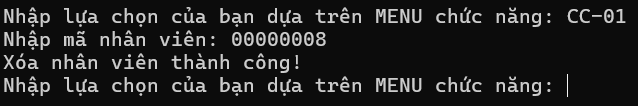
DD-01:Cập nhật Nhân viên



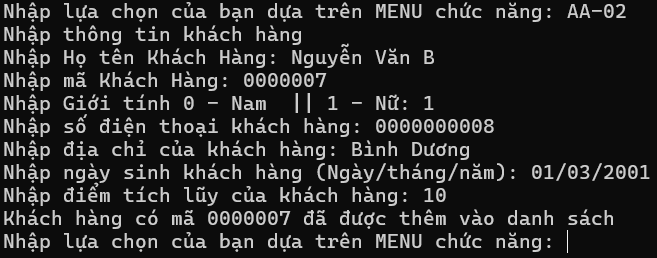
EE-01:In danh sách Mã nhân viên



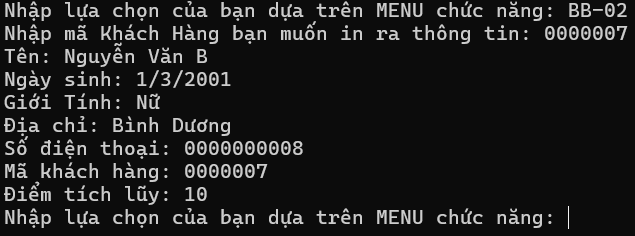
CC-01:Xóa thông tin Nhân Viên



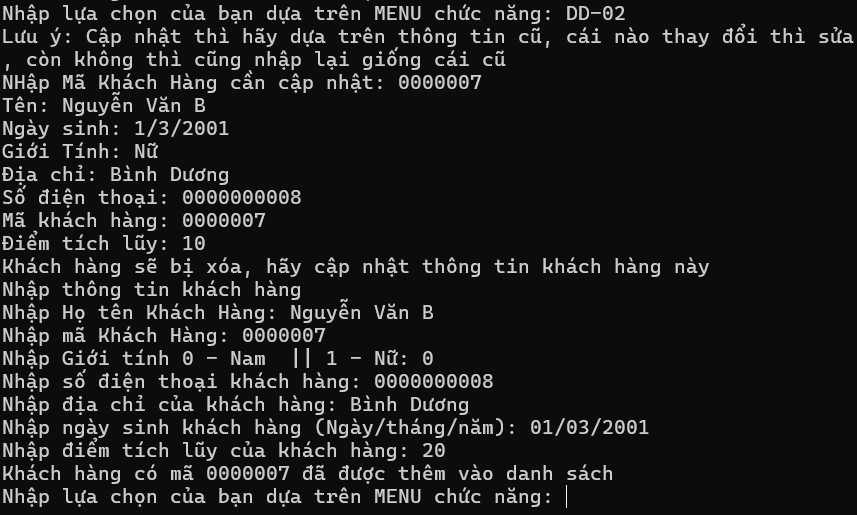
AA-02:Nhập thông tin và thêm Khách hàng mới



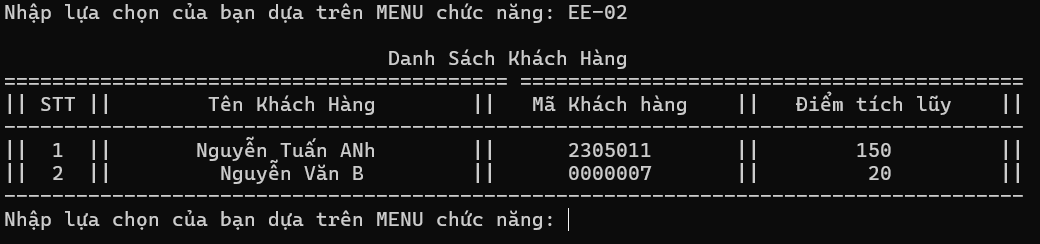
BB-02:Hiển thị Khách hàng



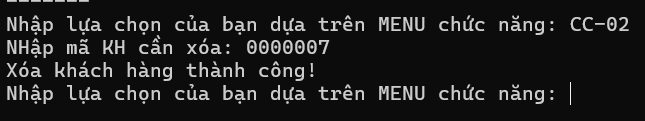
DD-02:Cập nhật Khách hàng



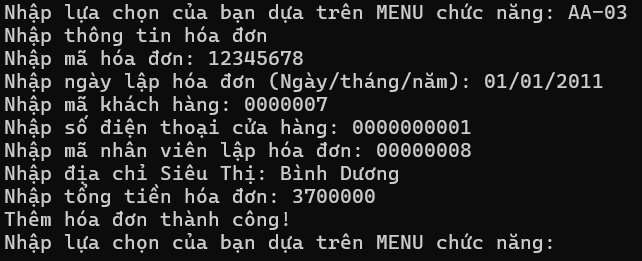
EE-02:In danh sách Mã khách hàng



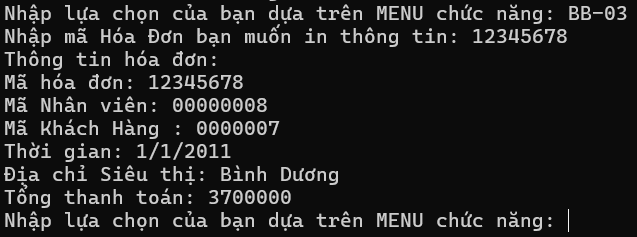
CC-02:Xóa thông tin Khách hàng



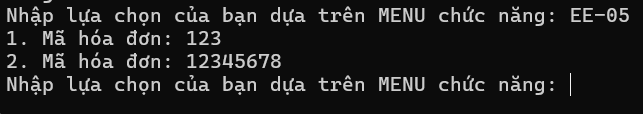
AA-03:Nhập thông tin và thêm Hóa đơn mới



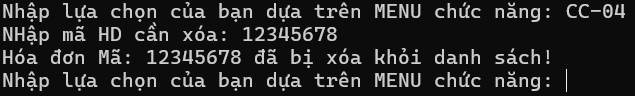
BB-03:Hiển thị thông tin Hóa đơn



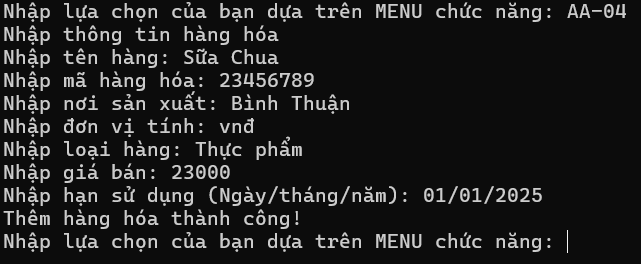
EE-05:In danh sách Mã hóa đơn



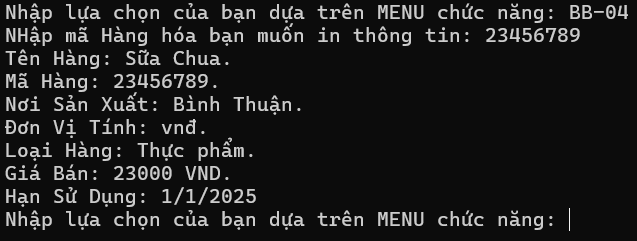
CC-04:Xóa thông tin Hóa đơn



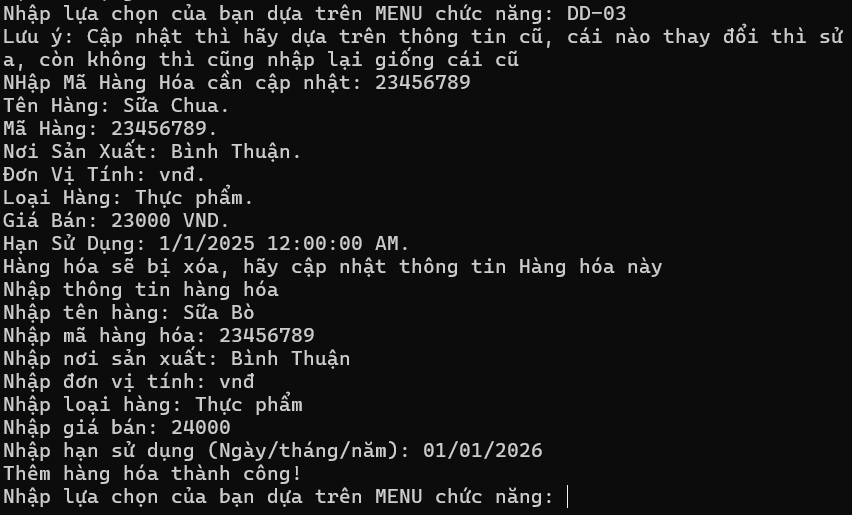
AA-04:Nhập thông tin và thêm Hàng hóa mới



BB-04:Hiển thị thông tin Hàng hóa



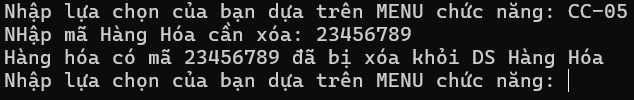
DD-03:Cập nhật Hàng hóa



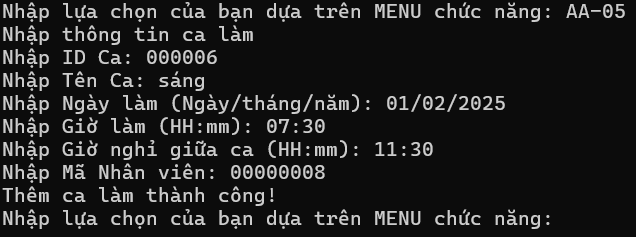
EE-03:In danh sách Mã hàng hóa



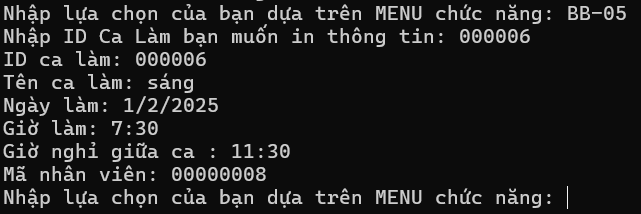
CC-05:Xóa thông tin Hàng hóa



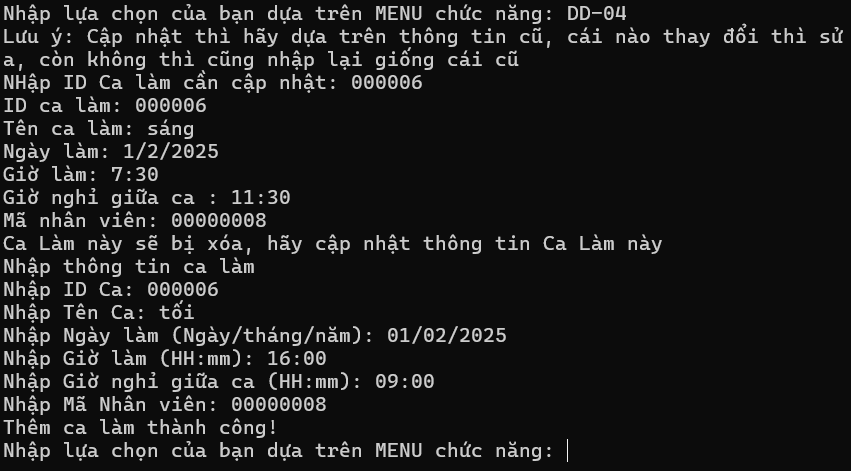
AA-05:Nhập thông tin và thêm Ca làm mới



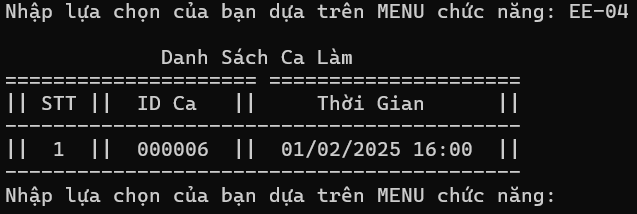
BB-05:Hiển thị thông tin Ca làm



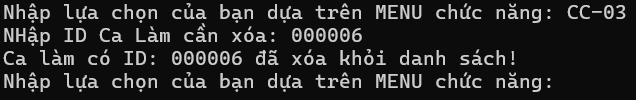
DD-04:Cập nhật Ca làm



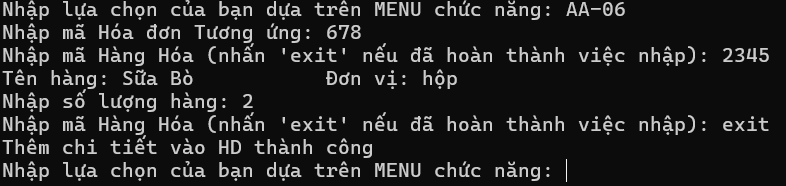
EE-04:In danh sách ID ca làm



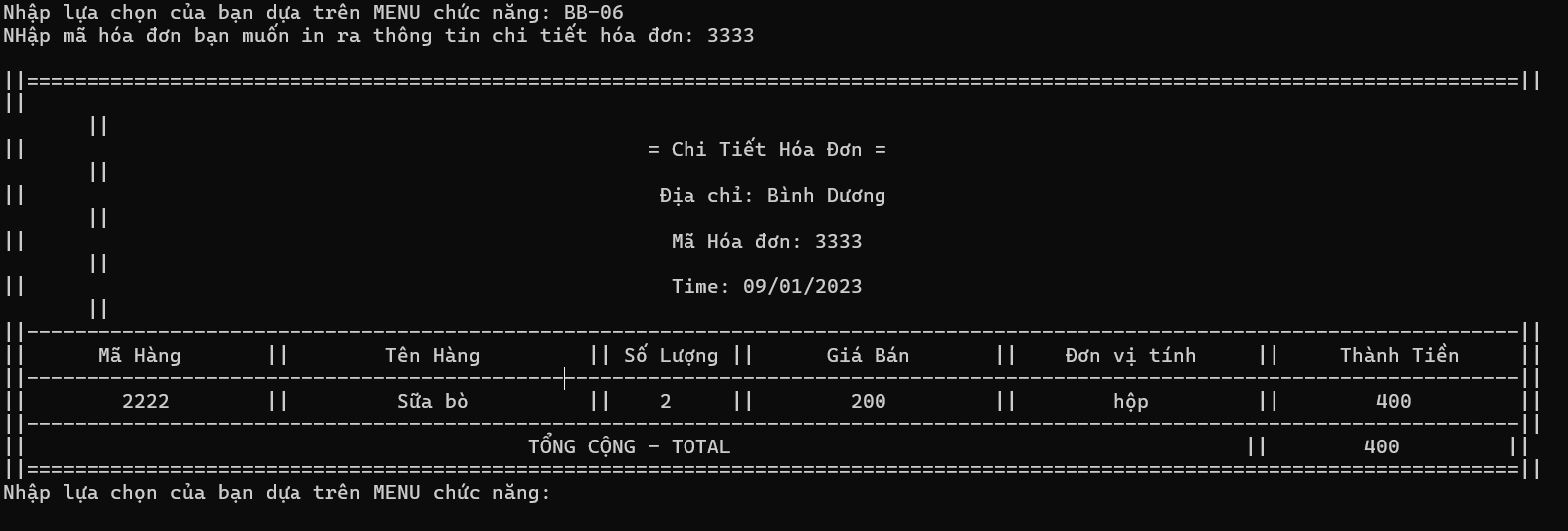
CC-03:Xóa thông tin Ca làm



AA-06:Nhập thông tin và thêm Chi tiết HD theo Mã HD:



BB-06:Hiển thị thông tin Chi tiết HD



DD-00:Thoát chương trình



# Kết Luận – Kiến nghị

## Đánh giá kết quả đạt được của đề tài so với yêu cầu của đề tài

Phân tích xác định thuật toán:

* Nhóm chúng em đã liệt kê ra các đối tượng trong đề tài “Lập trình quản lý hoạt động mua bán ở một siêu thị” (Bao gồm: Nhân viên, Khách hàng, Hàng hóa, Ca làm, Hóa đơn, Chi tiết hóa đơn) và trừu tượng hóa chúng thành các class tương ứng, mỗi class đều có các thuộc tính và phương thức mà nhóm chúng em cho là đã phù hợp với đề tài.
* Nhận diện các mối quan hệ: Trong bài sử dụng bao gồm Generalization, Association, Composition. Chúng được xác định và liên kết với nhau theo một sơ đồ, cũng như thể hiện rõ các Multiplicity giữa các mối quan hệ.
* Hai class “Nhân Viên” và “Khách Hàng” được tổng quát hóa lại bởi “Con Người” (Sử dụng Generalization).
* Sơ đồ mô hình hóa được vẽ ra thông qua IBM Rational Rose.

Class trung tâm:

Class “Hóa Đơn” là class trung tâm vì nó tạo ra nhiều mối quan hệ với các class khác trong quá trình hoạt động của đề tài.

Các chức năng của chương trình:

* Khi khởi động chương trình, thì sẽ tạo ra một menu thể hiện các chức năng của chương trình.
* Các chức năng chính của chương trình đó chính là việc xem, thêm, sửa, xóa.
* Ngoài ra còn có một số chức năng phụ, điển hình như việc tự động tính toán số tiền khi đã nhập các số liệu cần thiết.
* Cuối cùng là việc thoát ra khỏi các chức năng trong chương trình, cũng như tắt chương trình.

Lập trình:

* Các thuộc tính hầu như được đóng gói lại từ khai báo Private (trừ class “Con người” vì là class cha nên khai báo Protected).
* Về Method:
* Hàm khởi tạo sẽ mang tên của class.
* Các hàm đọc dữ liệu (get) và ghi dữ liệu (set).
* Hàm Info() thể hiện thông tin mô tả đối tương.
* Các hàm tính toán-xử lý:
* Có các hàm để kiểm tra xem việc nhập có đúng với các yêu cầu được đặt ra hay không.
* Việc tính toán chỉ cần nhập số liệu, chương trình sẽ xử lí phần còn lại.
* Mỗi hàm đều đã xác định kieru dữ liệu trả về tham số.
* Trong chương trình có áp dụng tính trừu tượng, phương pháp đóng gói, kế thừa,…

Ngôn ngữ lập trình:

* Sử dụng ngôn ngữ C#.
* Tuân theo các quy tắt cơ bản trong lập trình của C#, như quy tắc đặt tên, mỗi file.cs biểu thị cho một class,…

Phương pháp lập trình:

* Như yêu cầu của giảng viên, đoạn code được thiết kế theo phương pháp lập trình hướng đối tượng, bao gồm sự tham gia của class Program, class Control, các class Entity,…
* Tính hợp lý nhất quán:
* Các đoạn code và các class trong chương trình nhất quán với mô hình IBM Rational Rose.
* Đã kiểm tra đảm bảo về mối quan hệ của các class.

## Đề xuất xu hướng phát triển của đề tài

* Tinh chỉnh các đoạn code trở nên ngắn gọn hơn, tinh giảm độ phức tạp của thuật toán.
* Áp dụng thực tế nhiều hơn, tăng cường các điều kiện khiến chương trình trở nên thiết thực hơn.
* Thiết kế giao diện tiếp cận với người dùng hơn.
* Thay đổi để đơn giản hóa một số công đoạn nhập lệnh sang sử dụng click chuột, để người dùng có thể sử dụng một cách dễ dàng hơn.
* Đối với các loại mã cũng cần tinh chỉnh để phù hợp đối với thực tế.

**DANH MỤC TÀI LIỆU THAM KHẢO**