**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN LẬP TRÌNH**

**Đề tài:**

**ỨNG DỤNG QUẢN LÝ NHÀ HÀNG**

**SINH VIÊN THỰC HIỆN:**

**Trần Ngọc Huy**

**LỚP: 18TCLC-DT3 NHÓM: 18N15-N13**

**Nguyễn Lê Tuấn Cương**

**LỚP: 18TCLC-DT3 NHÓM: 18N15-N13**

**Hoàng Nguyên Vũ**

**LỚP: 18TCLC-DT3 NHÓM: 18N15-N13**

**Đà Nẵng 12-2019**

LỜI MỞ ĐẦU

1. **Lý do chọn đề tài:**

Công việc quản lý hệ thống là một công việc đòi hỏi độ chính xác cao, tối ưa và rõ ràng. Khi mà công nghệ chưa phát triển thì việc quản lý còn phải được thực hiện một cách thủ công và phải làm trên giấy, gây bất tiện mỗi khi cập nhật, thêm hay sửa xóa dữ liệu dẫn đến nhiều vấn đề tồn đọng trong khâu quản lý. Cho đến nay, khi mà công nghệ đã phát triển thì hàng loạt giải pháp tối ưu cho việc quản lý hệ thống đã ra đời, từ việc quản lý dữ liệu trên giấy tờ thì nay đã chuyển qua quản lý trên máy tính, thông qua các phần mềm, các dich vụ lưu trữ. Để tiếp cận với giải pháp trên, trong phạm vi trường học và phạm vi của đồ án, chúng em tiến hành xây dựng một phần mềm Quản lý nhà hàng nhằm tìm hiểu cách thức hoạt động của một hệ thống quản lý thông qua phần mềm trên máy tính, tìm hiểu quá trình kết nói cấu trúc thuật toán với cơ sở dữ liệu, qua đó xây dựng các tính năng mà một hệ thống quản lý cần có.

1. **Mục tiêu:**

* Mục tiêu chính của đồ án:

Hiểu và vận dụng được các kiến thức của các môn Kỹ thuật lập trình, Lập trình hướng đối tượng, cơ sở dữ liệu… Đồng thời nâng cao khả năng tự tìm tòi các kiến thức liên quan đến cấu trúc giải thuật và khả năng làm việc nhóm.MỤC LỤC

[LỜI MỞ ĐẦU 2](#_Toc27732970)

[MỤC LỤC 3](#_Toc27732971)

[DANH MỤC HÌNH VẼ 4](#_Toc27732972)

[1. GIỚI THIỆU ĐỀ TÀI 5](#_Toc27732973)

[2. CƠ SỞ LÝ THUYẾT 5](#_Toc27732974)

[2.1. Ý tưởng 5](#_Toc27732975)

[2.2. Cơ sở lý thuyết 5](#_Toc27732976)

[3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN 6](#_Toc27732977)

[3.1. Phát biểu bài toán 6](#_Toc27732978)

[3.2. Cấu trúc dữ liệu 6](#_Toc27732979)

[3.2.1. Mô tả 6](#_Toc27732980)

[3.2.2. Lớp Node 6](#_Toc27732981)

[3.2.3. Lớp Array 6](#_Toc27732982)

[3.3. Thuật toán 8](#_Toc27732983)

[4. CHƯƠNG TRÌNH VÀ KẾT QUẢ 9](#_Toc27732984)

[4.1. Tổ chức chương trình 9](#_Toc27732985)

[4.2. Ngôn ngữ cài đặt 9](#_Toc27732986)

[4.3. Kết quả 9](#_Toc27732987)

[4.3.1. Giao diện chính của chương trình 9](#_Toc27732988)

[4.3.2. Kết quả thực thi của chương trình 9](#_Toc27732989)

[4.3.3. Nhận xét 12](#_Toc27732990)

[5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 13](#_Toc27732991)

[5.1. Kết luận 13](#_Toc27732992)

[5.2. Hướng phát triển 13](#_Toc27732993)

[TÀI LIỆU THAM KHẢO 14](#_Toc27732994)

DANH MỤC HÌNH VẼ

[Hình 1](#_Toc27732586)

[Hình 2](#_Toc27732587)

[Hình 3](#_Toc27732588)

[Hình 4](#_Toc27732589)

[Hình 5](#_Toc27732590)

[Hình 6](#_Toc27732591)

[Hình 7](#_Toc27732592)

# GIỚI THIỆU ĐỀ TÀI

*Đề tài: Ứng dụng quản lý nhà hàng*

- Trong thực tế, khi mở ra một dịch vụ kinh doanh thì việc cấp thiết nhất là phải đề ra được một phương án quản lý dữ liệu hiệu quả, tối ưu và đem lại trực quan cho người dùng. Cụ thể ở đây là ứng dụng quản lý nhà hàng, nhóm chúng em đã vận dụng cá kiến thức có được ở lớp học và tìm hiểu thêm, đã hình thành được ứng dụng. Dù hiện tại chỉ đang được thực hiện thông qua giao diện console nhưng tương lai sẽ có thể phát triển cao hơn.

# NGHIỆP VỤ CHUNG VÀ CÁC CHỨC NĂNG

## Nghiệp vụ chung

- Quản lý các đối tượng cơ bản của nhà hàng để thiết lập hóa đơn và thanh toán trên cửa sổ console.

- Lưu dữ liệu vào hệ thống cơ sở dữ liệu của SQL Server.

## Chức năng

# 2.2.1. Quản lý món (Menu)

- Thuộc tính:

**Id** **Món**(FoodId)

Kiểu số nguyên.

**Tên** (Name)

Chỉ gồm các kí tự có trong bản chữ cái và các chữ số.

**Giá** (Cost)

Kiểu số thực. Đơn vị tính bằng USD.

- Tính năng

Thêm

Nhập **Tên** và **Giá** để thêm.

Hiển thị

Hiển thị tất cả các món có trong Menu.

Xóa

Nhập **Id Món** để xóa.

# 2.2.2. Quản lý bàn (Desk)

- Thuộc tính:

**Id** **Bàn**(DeskId)

Kiểu số nguyên.

**Khả dụng** (Available)

Kiểu đúng/sai. Chỉ trạng thái khả dụng của bàn.

- Tính năng

Thêm

Nhập **Id Bàn** để thêm.

Bàn sau khi thêm thì thuộc tính Khả dụng mặc định là đúng.

Hiển thị

Hiển thị tất cả các bàn.

# 2.2.3. Quản lý Hóa đơn (Bill)

- Thuộc tính:

**Id** **Hóa đơn**(BillId)

Kiểu số nguyên.

**Id Bàn** (DeskId)

Kiểu số nguyên và là giá trị của một Bàn tồn tại.

**Đã thanh toán** (Payment)

Kiểu đúng/sai. Hiển thị trạng thái thanh toán của bàn.

**Tổng** (Total)

Kiểu số thực. Là tổng thanh toán của Hóa đơn.

- Tính năng

Thêm

Hiển thị các Bàn còn trống.

Yêu cầu người dùng nhập Id Bàn cần tạo hóa đơn.

Hóa đơn sau khi tạo thì thuộc tính Đã thanh toán mặc định là sai.

Cập nhật Trạng thái thanh toán

Hiển thị thông tin các Hóa đơn chưa thanh toán.

Yêu cầu người dùng nhập Id Hóa đơn để xem thông tin chi tiết.

Hiển thị chi tiết hóa đơn.

Nhập Trạng thái thanh toán để cập nhật.

Sau khi thanh toán. Thuộc tính Khả dụng của Bàn tương ứng sẽ đặt về đúng.

# 2.2.4. Quản lý Gọi món (AddFood)

- Thuộc tính:

**Id Gọi Món** (AddFoodId)

Kiểu số nguyên

**Id Món** (FoodId)

Kiểu số nguyên.

**Id Hóa đơn** (BillId)

Kiểu số nguyên.

- Tính năng

Thêm

Hiển thị danh sách Hóa đơn chưa thanh toán.

Yêu cầu người dùng nhập Id Hóa đơn.

Hiển thị danh sách Món.

Yêu cầu người dùng nhập Id Món.

Yêu cầu người dùng nhập Số lượng.

Sau khi thêm Món. Tổng của Hóa đơn được tính lại.

3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

## Phát biểu bài toán

* Input: Dãy số P là (n+1) hệ số của đa thức P(x).

Dãy số Q là (m+1) hệ số của đa thức Q(x).

* Output: Dãy số R là (m+n) hệ số của đa thức R(x) = P(x) \* Q(x).

## Cấu trúc dữ liệu

### Mô tả

Gồm các Node được nối với nhau thông qua các liên kết. Trong đó có các hoạt động thêm, sửa, xóa, … được thực hiện bởi lớp Array.

### Lớp Node

1. Thành viên pubic

* *Hàm dựng*:

Khởi tạo giá trị ban đầu và gán liên kết tiếp theo (nexts) là rỗng.

* *Hàm lấy dữ liệu*:

lấy dữ liệu của Node. Trả về kiểu double.

1. Thành viên private

* *Dữ liệu*:

Kiểu double.

* *Liên kết tiếp theo*:

Con trỏ kiểu Node.

### Lớp Array

1. **Thành viên public**

* *Biến lưu kích thước* danh sách kiểu int:

Để lưu số phần tử của danh sách liên kết.

* *Hàm dựng*:

Gán địa chỉ *Node đầu danh sách* và *Node cuối danh* sách bằng rỗng.

* *Hàm dựng* có *tham số* là kích thước là tham trị kiểu int:

Gán địa chỉ *Node đầu danh sách* và *Node cuối danh* sách bằng rỗng. Gọi *Hàm chèn vào cuối* cho đến khi *biến lưu kích thước* bằng với *kích thước* được truyền vào.

* *Hàm hủy*:

Duyệt và hủy vùng nhớ được cấp phát động cho Node.

* *Hàm* *Chèn vào cuối* với *tham chiếu truyền* *vào* là tham chiếu hằng kiểu double*:*

Tạo *Node mới* có giá trị là *tham số truyền vào* là tham trị kiểu int. Nếu *Hàm kiểm tra danh sách* trả về *true.* Tức là danh sách rỗng, thì gán địa chỉ *Node đầu danh sách* và *Node cuối danh sách* bằng địa chỉ *Node mới.* Còn không thì cho địa chỉ *Node cuối danh sách* chỉ đến địa chỉ *Node mới. G*án địa chỉ *Node cuối danh sách* bằng địa chỉ *Node mới.*

* *Hàm Toán tử =* với *tham chiếu truyền vào* là tham chiếu kiểu Array:

Duyệt *tham chiếu truyền vào*. Trong mỗi lần lặp, nếu *i* còn nhỏ hơn *Biến lưu kích thước* của vùng nhớ hiện tại thì gán *Hàm toán tử []* tại *i* bằng *tham chiếu truyền vào* tại *i.* Còn không thì gọi *Hàm chèn vào cuối* với *tham chiếu* là *tham chiếu truyền vào* tại *i.*

* *Hàm toán tử []* với *tham số truyền vào* là tham trị kiểu int, trả về kiểu tham chiếu double:

Tạo *biến tạm* là con trỏ kiểu Node lưu địa chỉ của vùng nhớ hiện tại. Trong khi *tham số truyền vào* khác 0 và *biến tạm* khác rỗng. Thì gán *biến tạm* bằng bằng *Liên kết tiếp theo*. Trừ *tham số truyền* một đơn vị. Sau khi lặp gọi *Hàm lấy dữ liệu* của *biến tạm* và trả về cho hàm.

* *Hàm toán tử \** với 2 *tham chiếu truyền vào* là tham chiếu hằng kiểu Array, trả về một tham trị kiểu Array:

Tạo *biến tạm* kiểu Array, thực hiện nhân *tham chiếu truyền vào* và gán cho *biến tạm* bằng thuật toán được mô tả phía dưới. Trả về cho hàm *biến tạm.*

* *Hàm toán tử <<* với *tham chiếu truyền vào* kiểu ostream và Array, trả về tham chiếu kiểu ostream:

Duyệt *tham chiếu Array được truyền vào* và xuất bằng *tham chiếu ostream được truyền vào.* Trả về *tham chiếu ostream được truyền vào* cho hàm.

* *Hàm toán tử >>* với *tham chiếu truyền vào* kiểu istream và Array, trả về tham chiếu kiểu istream:

Duyệt *tham chiếu Array được truyền vào* và nhập bằng *tham chiếu istream được truyền vào.* Trả về *tham chiếu istream được truyền vào* cho hàm.

* *Hàm toán tử <<* với *tham chiếu truyền vào* kiểu ofstream và Array, trả về tham chiếu kiểu ofstream:

Duyệt *tham chiếu Array được truyền vào* và xuất bằng *tham chiếu ofstream được truyền vào.* Trả về *tham chiếu ofstream được truyền vào* cho hàm.

* *Hàm toán tử >>* với *tham chiếu truyền vào* kiểu ifstream và Array, trả về tham chiếu kiểu ifstream:

Duyệt *tham chiếu Array được truyền vào* và nhập bằng *tham chiếu ifstream được truyền vào.* Trả về *tham chiếu ifstream được truyền vào* cho hàm.

1. Thành viên private

* *Node đầu danh sách*: Con trỏ kiểu Node lưu địa chỉ của Node đứng đầu danh sách.
* *Node cuối danh sách*: Con trỏ kiểu Node lưu địa chỉ của Node đứng cuối danh sách.
* *Hàm kiểm tra danh sách* có rỗng hay không: Nếu *Node đầu danh sách* là rỗng thì danh sách rỗng. Trả về kiểu bool.

## Thuật toán

Thuật toán nhân 2 đa thức:

* Khai báo 1 danh sách liên kết R chứa kết quả có độ dài là (m+n).
* Thuật toán:

for i = 0 🡪 length(P)

for j = 0 🡪 length(Q)

R[i + j] = R[i + j] + P[i] \* Q[j]

* Trả về kết quả R.

# TỔ CHỨC MÃ NGUỒN CỦA DỰ ÁN

## Khối Tables

Chứa các Class trong đó:

* Có thuộc tính giống với các bảng của Co sở dữ liệu.
* Hàm dựng để khởi tạo các giá trị mặc định.
* Hàm nhập để chỉ nhập những thuộc tính cần thiết.
* Hàm xuất để chuẩn hóa khoảng cách các thuộc tính được in ra.
* Hàm head để hiển thị Tiêu đề của bảng.

## Khối List

Chứa lớp List là một lớp template. Lớp List là danh sách liên kết đơn dùng để chứa các Node là các Class trong khói Tables, lớp List gồm những thao tác:

* Hàm dựng để khởi tạo danh sách trống.
* Nhận một Node truyền vào, tạo một Node mới đầu danh sách và sao chép giá trị vào.
* Nhận một Node truyền vào, tìm kiếm và cập nhật giá trị cho Node đó.
* Nhận một Node truyền vào, tìm kiếm và xóa Node đó.
* Xuất tất cả các Node của danh sách.
* Làm rỗng danh sách.
* Xem danh sách có rỗng hay không.

## Khối Views

Chịu trách nhiệm xuất ra Console và tương tác với người dùng. Nhận giá trị người dùng nhập vào và trả về cho tên hàm. Gồm các hàm với chức năng như sau:

Trang chủ

Chờ người dùng nhập một số để điều hướng tới phần quản lí các đối tượng: **Món, Bàn, Hóa Đơn, Gọi Món.**

**Món**

Chờ người dùng nhập số để điều hướng tới chức năng Thêm, Hiển Thị hoặc Xóa

Thêm Món

Nhận giá trị người dùng nhập vào và trả cho tên hàm.

Hiển thị Món

Hiển thị tất cả các món

Xóa Món

Nhận giá trị người dùng nhậpv à trả cho tên hàm.

**Bàn**

Chờ người dùng nhập số để điều hướng tới các chức năng Thêm hoặc Hiển thị.

Thêm Bàn

Nhận giá trị người dùng nhập và trả cho tên hàm.

Hiển thị bàn

Hiển thị tất cả các bàn.

**Hóa đơn**

Chờ người dùng nhấp số để điều hướng tới các chức năng Thêm hoặc Cập nhật trạng thái thanh toán.

Thêm Hóa đơn

Nhận giá trị người dùng nhập và trả cho tên hàm.

Cập nhật trạng thái hóa đơn

Hiển thị tất cả danh sách hóa đơn. Nhận giá trị người dùng nhập và trả cho tên hàm.

Chi tiết hóa đơn

Hiển thị tất cả danh sách món của bàn đó. Nhận giá trị người dùng nhập và trả cho tên hàm.

**Gọi món**

Chờ người dùng nhập số để điều hướng tới chức năng Thêm Gọi Món

Thêm Gọi món

Nhận giá trị người dùng nhập và trả cho tên hàm.

## Khối Models

Xử lý dữ liệu, đồng bộ giữa Cơ sở dữ liệu của SQK Server và Danh sách liên kết của khối List. Gồm các chức năng:

* Đồng bộ dữ liệu từ Cơ sở dữ liệu của SQL Server về Danh sách liên kết.
* Thêm, sửa, xóa dữ liệu trong Danh sách liên kết và sau đó đồng bộ Danh sách liên kết về Cơ sở dữ liệu của SQL Server.

## Khối Models

Tạo mạch liên kết qua lại giữa các khối Views. Đóng gói dữ liệu được trả về từ khối Views, sau đó gọi khối Models để xử lý. Chứa các hàm có tên giống với các hàm của khối Views. Những xử lý nhỏ có thể xử lý trực tiếp tại Hàm mà không cần gọi khối Models.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

* Nắm được lý thuyết và cách cài đặt danh sách liên kết đơn sử dụng ngôn ngữ lập trình C++
* Hiểu được lý thuyết và cách thực hiện đa năng hóa các toán tử **[] << >> \***
* Biết một số thao tác đọc dữ liệu từ file và ghi dữ liệu ra file.
* Hiểu và xây dựng thành công thuật toán nhân 2 đa thức.
* Nắm được các bước để xây dựng cấu trúc của một chương trình đơn giản.

## Hướng phát triển

* Tối ưu hóa chương trình bằng việc tối ưu hóa các hàm.
* Thêm các hàm xử lý ngoại lệ để hoàn thiện chương trình.
* Mở rộng chương trình với việc thực hiện các phép tính khác trên đa thức.

TÀI LIỆU THAM KHẢO

[1] GV. Lê Thị Mỹ Hạnh, Giáo trình môn học Lập trình hướng đối tượng, Đà Nẵng, 09/2002

[2] Dạy Nhau Học, Một số thao tác đọc dữ liệu từ File trong C++, https://cpp.daynhauhoc.com/10/2-mot-so-thao-tac-doc-du-lieu-tu-file-trong-c/, 17/11/2019

[3] Dạy Nhau Học, Một số thao tác ghi dữ liệu vào File trong C++, https://cpp.daynhauhoc.com/10/3-mot-so-thao-tac-ghi-du-lieu-vao-file-trong-c/, 17/11/2019

[4] List (abstract data type), https://en.wikipedia.org/wiki/List\_(abstract\_data\_type)