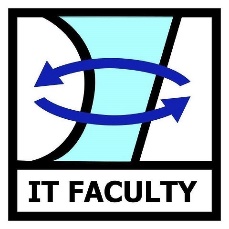
**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**---🙣\*🙡---**

****

**BÁO CÁO ĐỒ ÁN**

**PBL2: DỰ ÁN CƠ SỞ LẬP TRÌNH**

**HỆ THỐNG QUẢN LÝ SIÊU THỊ**

**(MINIMART MANAGEMENT SYSTEM)**

GIẢNG VIÊN HƯỚNG DẪN: **TS. ĐẶNG HOÀI PHƯƠNG**

SINH VIÊN THỰC HIỆN:

**LÊ VÕ NHẬT MINH LỚP: 22T\_DT4**

**NGUYỄN HỮU PHÚC LỚP: 22T\_DT4**

**NHÓM HỌC PHẦN: 22NH.13**

Đà Nẵng, ngày 23 tháng 12 năm 2023

**LỜI MỞ ĐẦU**

Trong thời đại công nghệ tiên tiến, công nghệ thông tin (IT) đã trở thành một ngành nghề có vai trò quan trọng đối với sự tồn tại và phát triển của đất nước. Với việc ứng dụng rộng rãi các thành tựu của ngành công nghệ thông tin vào nhiều lĩnh vực, việc giải quyết các vấn đề bằng cấu trúc dữ liệu và giải thuật luôn được các nhà phát triển đặc biệt quan tâm. Điều này đòi hỏi phải liên tục củng cố và cải tiến công nghệ, tối ưu các thuật toán để xây dựng thêm nhiều chức năng.

Với mong muốn xây dựng một phương pháp lập trình trực quan, mô tả trung thực các hệ thống trong thế giới thực, phương pháp  *“Lập trình hướng đối tượng”* đã ra đời. Lập trình hướng đối tượng là phương pháp lập trình xây dựng chương trình dựa trên đối tượng. Hơn nữa, một trong những nguyên tắc cơ bản quan trọng ảnh hưởng trực tiếp đến các thuật toán là *"Cấu trúc dữ liệu"*.

Nhằm nghiên cứu, xây dựng cấu trúc dữ liệu và giải thuật một cách hợp lý, đồng thời đào sâu, mở rộng kiến ​​thức về lập trình hướng đối tượng, nhóm chúng em đã chọn đề tài để nghiên cứu mang tên *“Hệ thống quản lý siêu thị”.*

Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến thầy Đặng Hoài Phương. Thầy đã tận tình giúp đỡ, hướng dẫn, giải đáp thắc mắc và chỉ ra những sai sót trong quá trình làm đồ án để chúng em có thể hoàn thành đồ án này. Thông qua dự án, chúng em đã học tập được rất nhiều kinh nghiệm quý báu từ thầy. Một lần nữa, chúng em xin được bày tỏ lòng biết ơn đến thầy Đặng Hoài Phương.

Kết quả đạt được của dự án này chắc hẳn không thể tránh khỏi những thiếu sót. Vì vậy, chúng em rất mong được thầy cô bổ sung thêm để dự án này được hoàn thiện hơn; đồng thời qua đó chúng em cũng rút ra nhiều điều cho các dự án sau này.

Chúng em xin chân thành cảm ơn!

Đà Nẵng, ngày 23 tháng 12 năm 2023

**Nhóm sinh viên thực hiện đề tài**

**1. Lê Võ Nhật Minh**

**2. Nguyễn Hữu Phúc**

**MỤC LỤC**

**DANH MỤC HÌNH VẼ**

# 

# 

# GIỚI THIỆU ĐỀ TÀI

## Đặt vấn đề

Bài toán quản lý luôn là vấn đề được quan tâm ở tất cả các lĩnh vực như quản lý nhân sự, quản lý kinh doanh, quản lý tài chính, …. Một trong số đó, vấn đề về giải pháp quản lý siêu thị cũng được đặt ra đối với người quản lý. Họ thường quản lý một cách thủ công. Với cách làm này, người quản lý không thể biết chắc chắn số lượng các khách hàng đã mua, số lượng mặt hàng đã bán ra, chi tiết hóa đơn với từng khách hàng, … Điều này dẫn tới những bất cập trong việc bảo mật thông tin, tính toán chi phí, doanh thu của siêu thị; người quản lý sẽ không biết đâu là các mặt hàng có mức độ tiêu thụ cao, không đánh giá được mức độ tồn kho của mặt hàng,…

Trong bối cảnh kinh doanh ngày càng phát triển và cạnh tranh khốc liệt, quản lý một cửa hàng kinh doanh như siêu thị nhỏ không phải là nhiệm vụ dễ dàng. Điều này đặt ra yêu cầu cần thiết phải có một hệ thống quản lý siêu thị hiệu quả, hỗ trợ trong việc tổ chức, theo dõi mọi hoạt động kinh doanh một cách minh bạch và tự động.

## Giới thiệu đề tài:

Việc tin học hóa quản lý siêu thị mang rất nhiều lợi ích so với các cách làm thủ công. Người điều hành sẽ quản lí siêu một cách chính xác và thuận tiện dựa trên những chức năng mà hệ thống đã được lập trình sẵn như kiểm tra tình trạng mặt hàng, lô hàng; quản lí thông tin của khách hàng, mặt hàng; tìm kiếm, kiểm tra thông tin của khách hàng, mặt hàng; thay đổi thông tin của khách hàng, mặt hàng; theo dõi dữ liệu bán hàng, bao gồm thông tin sản phẩm, số lượng bán ra, thu nhập từng ngày; quản lý hiệu quả tồn kho, cập nhật số lượng sản phẩm tự động, thông báo các mặt hàng sắp hết hoặc quá hạn sử dụng,… Với hệ thống này, việc quản lí sẽ vô cùng thuận lợi và dễ dàng, giúp tăng nâng suất cũng như hạn chế tối thiểu những sai sót trong công việc.

# PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

Hệ thống được viết thành một chương trình, người sử dụng là người quản lý siêu thị. Chương trình được thiết kế đáp ứng tương đối đủ các nhu cầu của người sử dụng

## 2.1. Định nghĩa và mô tả hệ thống

### 2.1.1. Các thực thể có tác động đến hệ thống

* Người quản lý siêu thị (Minimart manager): là người thực hiện công tác quản lý, trực tiếp thao tác với ứng dụng quản lý. Người quản lý nắm bắt được thông tin của lô hàng, mặt hàng được nhập vào kho, tình trạng của các mặt hàng, thông tin cá nhân của khách hàng, hóa đơn đơn của khách hàng, các hoạt động mua bán, tính toán hóa đơn, doanh thu của siêu thị.
* Mặt hàng (Products): là những vật được nhập từ lô hàng vào kho, được lưu trữ và quản lý trong siêu thị. Mỗi mặt hàng sẽ gắn với lô hàng, có mã (ID) cụ thể và duy nhất. Người quản lý còn quan tâm đến số lượng trong kho và hạn sử dụng của mặt hàng.
* Khách hàng (Customers): là những người đăng kí thông tin vào danh sách quản lý khách hàng của siêu thị, mỗi người khi đăng kí sẽ được cấp mã (ID). Thông tin của khách hàng sẽ được lưu trong hệ thống. Ngoài mã được cấp, khách hàng còn được định danh bằng số điện thoại.

### 2.1.2. Mô tả hoạt động của hệ thống

* Khi siêu thị nhập một lô hàng mới, người quản lý thực hiện nhập thông tin của các mặt hàng có trong lô vào hệ thống. Hệ thống sẽ kiểm tra danh sách và thông tin đã nhập có chính xác và tồn tại trong kho hay chưa. Nếu đã tồn tại thì cộng dồn số lượng của sản phẩm ở lô mới vào số lượng mặt hàng ở lô cũ, nhưng vẫn phân biệt theo mã của lô hàng. Ngược lại, thêm mặt hàng mới vào hệ thống.
* Khi khách hàng được thêm vào hệ thống, họ phải cung cấp thông tin cho người quản lý. Nếu thông tin của khách hàng chưa tồn tại trong hệ thống thì người quản lý sẽ tạo mới thông tin của khách hàng trong hệ thống. Ngược lại, nếu hệ thống đã có thông tin của khách hàng thì đưa ra thông báo đã tồn tại. Trường hợp thông tin mà khách hàng cung cấp không hợp lệ sẽ yêu cầu tái cung cấp thông tin.
* Trong quá trình vận hành, nếu có thông tin của mặt hàng, khách hàng và hóa đơn cần được tìm kiếm, chỉnh sửa, cập nhật, người quản lý siêu thị có thể thực hiện các thao tác trên trong hệ thống thông qua ID của các đối tượng.
* Khi khách hàng mua các mặt hàng của siêu thị, họ cần cung cấp thông tin cá nhân của mình và thông tin của mặt hàng (Tên mặt hàng, số lượng mua,…) cho người quản lý. Người quản lý thiết lập hóa đơn, lưu lại lịch sử hóa đơn đối với mỗi khách hàng, đồng thời khấu trừ số lượng mặt hàng đã mua vào hệ thống.

## 2.2. Mô hình nghiệp vụ

### 2.2.1. Sơ đồ phân rã chức năng

A diagram of a company

Description automatically generatedHình 2.1. Sơ đồ phân rã chức năng hệ thống

### 2.2.2. Mô tả chi tiết các chức năng

1.1. Thêm mặt hàng: Thêm thông tin của mặt hàng từ lô hàng hoặc nhập thông qua người quản lý.

1.2. Tìm kiếm mặt hàng: Cho phép người quản lý tìm kiếm mặt hàng theo mã (ID), tên mặt hàng, xuất xứ của mặt hàng đó.

1.3. Quản lý mặt hàng hết hạn sử dụng: Đưa ra danh sách các mặt hàng hết hạn sử dụng trong vòng x ngày tới (x là số ngày mà người quản lý muốn kiểm tra hạn sử dụng của mặt hàng). Ngoài ra còn cho phép người quản lý xóa mặt hàng hết hạn/ sắp hết hạn ra khỏi hệ thống.

2.1. Thêm khách hàng mới: Thêm thông tin của khách hàng từ một danh sách có sẵn hoặc được nhập bởi người quản lý.

2.2. Tìm kiếm khách hàng: Cho phép tìm kiếm khách hàng trong hệ thống thông qua số điện thoại hoặc tên khách hàng, dữ liệu trả về là danh sách khách hàng cần tìm kiếm, đồng thời cho phép xem lịch sử hóa đơn của mỗi khách hàng tìm

kiếm được.

2.3. Chỉnh sửa thông tin của khách hàng: Nếu khách hàng có nhu cầu thay đổi thông tin cá nhân, thay đổi thông tin của mỗi khách hàng bằng số điện thoại và cập nhật lại các chi tiết hóa đơn của khách hàng đó trong hệ thống.

2.4. Xóa khách hàng: Cho phép người quản lý xóa khách hàng ra khỏi hệ thống nếu khách hàng không mua hàng trong thời gian dài.

2.5. Hiển thị danh sách khách hàng: Hiển thị danh sách các khách hàng theo số điện thoại hoặc tên khách hàng như trong chức năng tìm kiếm, ngoài ra còn có thể hiển thị toàn bộ danh sách khách hàng.

3.1. Tính hóa đơn: Tạo hóa đơn bao gồm có thông tin của người mua hàng, thông tin mặt hàng được mua, thời gian tạo hóa đơn và lưu lại trong hệ thống.

3.2. Tính toán hóa đơn: Từ các chi tiết hóa đơn, người quản lý có thể tính tổng doanh thu mua hàng trong các khoảng thời gian xác định.

3.3. Tìm kiếm hóa đơn: Một khách hàng có thể có nhiều hóa đơn, người quản lý có thể tìm kiếm hóa đơn dựa vào mã hóa đơn, số điện thoại của người mua hoặc ngày tạo ra hóa đơn.

3.4. Hiển thị danh sách hóa đơn của khách hàng: Cho phép xem toàn bộ hóa đơn cũng như chi tiết hóa đơn của mỗi khách hàng trong hệ thống.

# 3. THIẾT KẾ CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

Như đã trình bày ở, yêu cầu chính của phần mềm chính là quản lý các đối tượng thuộc 3 lớp Product, Customer và Bill. Nói cách khác, trong suốt quá trình mà phần mềm chạy, nó sẽ lưu trữ và thực hiện các thao tác đối với 3 danh sách, mỗi danh sách sẽ lưu trữ các đối tượng của 1 trong 3 lớp. Để thực thi các danh sách này trong phần mềm, chúng tôi đã sử dụng danh sách liên kết đôi (double-linked list).

## 3.1. Phát biểu bài toán

Chủ của một MiniMart mong muốn sẽ có một phần mềm có thể hỗ trợ trong việc quản lý siêu thị của mình. Phần mềm gồm 2 chức năng cơ bản, bao gồm: tính hóa đơn cho khách hàng và thực hiện việc quản lý.

Đối với chức năng đầu tiên, yêu cầu phần mềm có thể tính toán và xuất ra được hóa đơn với số tiền mà khách hàng cần phải thanh toán một cách chính xác.

Đối với chức năng thứ hai, việc quản lý ở đây sẽ được thực hiện đối với các đối tương thuộc vào 3 lớp, là: sản phẩm (Product), khách hàng (Customer) và hóa đơn (Bill). Mỗi lớp bên sẽ tương ứng với 1 bảng trong CSDL và có các trường (Field) sau:

* Product: P\_ID (PK), Name, Quantity/Weight, Price, Origin.
* Customer: C\_ID (PK), Phone , Name, Address.
* Bill: B\_ID (PK), C\_ID, Buy datetime, Total.

Ngoài ra, còn có bảng chi tiết sản phẩm (Detail Product) lưu trữ thông tin chi tiết về sản phẩm trong mỗi lần nhập hàng (như HSD, NSX của sản phẩm trong đợt nhập, ngày nhập hàng, …) và chi tiết hóa đơn (Detail Bill) lưu trữ thông tin chi tiết về từng mặt hàng mà khách hàng mua trong hóa đơn (như số lượng sản phẩm được mua, …)

* Detail Product: DP\_ID (PK), P\_ID, Quantity/Weight, EXP (hạn sử dụng) , MFG (ngày sản xuất), RECIVE DAY (ngày nhập sản phẩm).
* Detail Bill: B\_ID (PK), DP\_ID (PK), Quantity/Weight (số lượng mỗi sản phẩm mà khách hàng mua.

Như vậy, bài toán mà ta cần giải quyết đó chính là tổ chức và quản lý CSDL gồm 5 bảng – Product, Detail Product, Customer, Bill và Detail Bill.

## 3.2. Phân tích và ứng dụng cấu trúc dữ liệu và thuật toán trong hệ thống

Như đã trình bày ở, yêu cầu chính của phần mềm chính là quản lý các đối tượng thuộc 3 lớp Product, Customer và Bill. Nói cách khác, trong suốt quá trình mà phần mềm chạy, nó sẽ lưu trữ và thực hiện các thao tác đối với 3 danh sách, mỗi danh sách sẽ lưu trữ các đối tượng của 1 trong 3 lớp. Để thực thi các danh sách này trong phần mềm, chúng tôi đã sử dụng danh sách liên kết đôi (double-linked list).

### Danh sách liên kết đôi

Danh sách liên kết đôi gồm các nút (node) liên với nhau theo 2 chiều. Mỗi node gồm 3 trường:

* Data: chứa dữ liệu của phần tử trong mỗi của danh sách.
* Pre: chứa liên kết (con trỏ) tới nút liền trước nó trong danh sách. Trong trường hợp là nút đầu tiên (không có nút liền trước), trường này được gán 1 giá trị đặc biệt (NULL).
* Next: chứa liên kết (con trỏ) tới nút kế tiếp (liền sau) nó trong danh sách. Trong trường hợp là nút cuối cùng (không có nút liền sau), trường này được gán 1 giá trị đặc biệt (NULL).

### Thực thi trong phần mềm

Trong phần mềm, danh sách liên kết đôi được cụ thể hóa dưới dạng class template DList gồm 2 thuộc tính Head, Tail lần lượt chứa liên kết (con trỏ) tới nút đầu tiên và nút cuối cùng trong danh sách.

Node trong danh sách liên kết cũng sẽ được xây dựng một lớp tương ứng – class template DNode gồm 3 thuộc tính: pre, data và next.

## 3.3. Thuật toán

Để hỗ trợ việc quản lý một cách hiệu quả thì việc tìm kiếm một hoặc nhiều đối tượng thỏa một điều kiện nào đó cũng như việc sắp xếp danh sách theo tiêu chí nào đó là điều tất yếu. Vì vậy, trong phần mềm, chúng tôi đã sử dụng thuật toán tìm kiếm tuần tự (linear serch) và thuật toán quich sort để thực 2 công việc trên.

Thông thường, 2 thuật toán này được sử dụng trên cấu trúc dữ liệu mảng như trong phần mềm chúng tôi sử dụng danh sách liên kết đôi nên trong phần trình bày sau đây, chúng tôi sẽ mô tả thuật toán bằng danh sách liên kết đôi.

### Tìm kiếm tuần tự (Linear search)

Với thuật toán này, từng phần tử trong danh sách sẽ lần lượt được duyệt, bắt đầu từ phần tử đầu, cho đến khi tìm được phần tử mong muốn hoặc duyệt tới phần tử cuối cùng của danh sách (trong trường hợp không tìm thấy).

Input: Danh sách cần thực hiện tìm kiếm, giá trị cần tìm kiếm

Output: phần tử chứa giá trị cần tìm hoặc NULL nếu không tìm thấy

Thuật toán

B1: Khởi tạo 1 con trỏ curPtr và gán cho nó giá trị Head của danh sách

B2: Nếu giá trị của nút mà curPtr đang trỏ tới bằng với giá trị cần tìm hoặc   
curPtr = NULL thì chuyển tới B4

B3: curPtr = curPtr→next. Quay lại B2.

B4: Trả về kết quả

Mã giả của thuật toán trên như sau:

LINEAR\_SEARCH (DList ls, DATA\_TYPE data\_lookup)

1. DATA\_TYPE \* curPtr = ls.Head
2. While (curPtr ≠ NULL) AND (curPtr→data ≠ data\_lookup) do
3. curPtr = curPtr→next
4. Return curPtr

Đánh giá thuật toán

* Độ phức tạp thuật toán trong tình huống xấu nhất là O(n) khi phần tử tìm kiếm nằm cuối danh sách hoặc không có trong danh sách.
* Thời gian chạy tốt nhất: O(1) khi phần tử cần tìm nằm ngay đầu danh sách.
* Thời gian chạy trung bình: O(n)

### 3.3.2. Quick sort

Đây là thuật toán sắp xếp dựa trên thuật toán chia để trị (Divide and conquer). Thuật toán này sẽ chọn 1 phần tử làm pivot and phân mảng ban đầu thành 2 phần trước và sau pivot được chọn (partition) bằng cách đặt pivot vào đúng vị trí của nó trên mảng ban đầu khi nó đã được sắp xếp.

***Thuật toán partition***

Có nhiều sự hướng để chúng ta chọn pivot. Ở đây, chúng tôi sẽ sử dụng cách luôn lấy phần tử cuối cùng làm pivot. Khác với khi ta sử dụng thuật toán cho mảng, ở đây pivot là 1 còn trỏ giữ địa chỉ của node cuối cùng trong danh sách

Ý tưởng của thuật toán khá đơn giản. Chúng ta sẽ bắt đầu duyệt từ đầu đến cuối của danh sách, phần đầu tiên trong dánh sách và sử dụng con trỏ bound lưu giữ vị trí của of các phần tử (địa chỉ của node) nhỏ hơn hoặc bằng giá trị node mà pivot đang trỏ tới. Trong quá trình duyệt, nếu giá trị của phần tử nhỏ hơn hoặc bằng pivot→data thì ta tịnh tiến bound lên 1 phần tử và đổi giá trị của phần tử đó cho bound →data.

Input: Danh sách có địa chỉ node đầu là Left, node cuối là Right

Output: Vị trí của pivot(địa chỉ nút chọn làm pivot) trong danh sách khi nó được sắp xếp

B1: pivot = Right, curPtr = Left, bound=NULL

B2: Nếu curPtr = Right thì chuyển tới B5

B3: Nếu curPtr→data <= pivot→data thì

bound = bound →next

Swap (bound, curPtr)

B4: curPtr = curPtr→next. Quay lại B2

B5: Swap(bound→next, pivot) và Trả về bound→next

Thời gian chạy O(n)

***Thuật toán quicksort***

Input: Danh sách có địa chỉ node đầu là Left, node cuối là Right

Output: Danh sách đã sắp xếp

B1: Nếu Left = NULL hoặc Right= NULL hoặc Left=Right hoặc Left→pre=Right hoặc Right →next=Left thì chuyển tới B5

B2: Gán cho con trỏ p = Partition(Left, Right)

B3: Gọi đệ quy Quicksort(Left, p →pre)

B4: Gọi đệ quy Quicksort(p →next, Right)

B5: Kết thúc

Đánh giá thuật toán

* Độ phức tạp thuật toán trong tình huống xấu nhất là O(n2) khi danh sách đã được sắp xếp theo đúng thứ tự
* Thời gian chạy tốt nhất: O(nlgn) khi pivot được chọn luôn chia mảng thàng 2 phần có lượng giống như
* Thời gian chạy trung bình: O(nlgn)

# 4. PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI

# HỆ THỐNG

## 4.1. Cấu trúc hệ thống hướng đối tượng (class, object, relation)

Để có thể mô hình các đối tượng cần quản lý trên thực tế thành các cấu trúc trong chương trình một cách trực quan, lập trình hướng đối (Oject Oriented Programming hay OOP) chính là phương pháp hiệu quả nhất. Phần mềm “Quản lý siêu thị” sử dụng phương pháp này để có thể đơn giản hóa việc quản lý các đối tượng, cũng như mối quan hệ giữa chúng. Vậy các lớp (Class), đối tượng (Object) và mối quan hệ (Relation) giữa chúng được xây dựng như thế nào trong phần mềm của chúng tôi ? Điều đó sẽ được trình bày ngay sau đây.

### Lớp (Class)

Mỗi lớp sẽ chứa các thuộc tính đặc trưng với quyền truy cập private đảm bào tính đóng gói của OOP.

Bên cạnh đó, các lớp cũng sẽ bao gồm các hàm khởi tạo (constructor), hàm hủy (destructor). Việc truy cập dữ liệu (các thuộc tính trong class) được thực hiện thông qua các cặp hàm Get-Set tương ứng. Ngoài ra, một số lớp còn có những hàm hỗ trợ cho quản lý, xử lý dữ liệu cho từng lớp cụ thể.

#### 4.1.1.1 Lớp Product

Lớp này gồm các thuộc tính với quyền truy cập pivate:

* + ID: String
  + Name: String
  + NumDP: int (số lần nhập hàng)
  + Price: int
  + SL: int (Tổng số lượng sản phẩm hiện có của tất cả các đợt nhập hàng)
  + XS: String (Xuất xứ)
  + isDeleted: bool (dùng để xóa mềm sản phẩm cho biết sản phẩm đã bị xóa hay chưa)
  + List: DList<DetailProduct \*> (Danh sách lưu thông tin chi tiết sản phẩm mỗi đợt nhập hàng)

Và các phương thức với quyền truy cập public:

* + Product(const String&, const String&, const String&, cons tint&)
  + ~Product()
  + Các cặp hàm Get-Set cho mỗi thuộc tính
  + void AddDetailProduct(DetailProduct \*) : thêm 1 con trỏ tới 1 đối tượng DetailProduct vào danh sách List
  + void DeleteDP(DetailProduct \*ptrDP) : xóa đối tượng DetailProduct trỏ bỏi con trỏ ptrDP ra khỏi List
  + DetailProduct\* FindDetailProduct(const String& id) const : thực hiện tìm kiếm và trả lại 1 con trỏ tới đối tượng DetailProduct có ID là id
  + void Deduct (const int sl) và void Add(const int sl) : giảm và tăng thuộc tính SL một lượng bằng sl
  + bool SLWarn() const : trả về true nếu số lượng sản phẩm <=10, ngược lại, trả về fasle

#### 4.1.1.2. Lớp DetailProduct và lớp DetailWeaightProduct

Lớp Product chỉ lưu các thông tin chung nhất về nhất sản phẩm. Tuy nhiên, mỗi sản phẩm sẽ có nhiều đợt nhập hàng. Mỗi đợt nhập hàng lại có sự khác nhau về ngày sản xuất, hạn sử dụng, ngày nhập, số lượng sản phẩm được nhập của đợt hàng đó. Điều này yêu cầu cần có 1 lớp để mô phỏng các đối tượng này – lớp DetailProduct.

Ngoài ra, mỗi sản phẩm trong siêu thị sẽ gồm 2 loại chính: một loại sẽ tính theo từng sản phẩm một và một loại sẽ tính theo cân(ví dụ như trái cây, thịt,…), do đó cần có 2 lớp: DetailProduct và DetailWeightProduct. Nhìn chung, 2 lớp này sẽ có các thuộc giống nhau. Tuy nhiên, chúng lại khác nhau về cách tính giá cho sản phẩm, điều kiện kiểm tra gần hết hàng hay chưa và lớp các sản phẩm sẽ có thêm thuộc tính đơn vị tính cho giá tiền (giá tiền đó là của 100g hay 1kg). Vì vậy, ta sẽ xây dựng lớp DetailWeightProduct kế thừa từ lớp DetailProduct và để nó override lại những phương thức cần thiết.

* Lớp DetailProduct

Lớp này gồm các thuộc tính với quyền truy cập pivate:

* ID: string
* NN: Date (ngày nhập của đợt hàng)
* NSX: Date (ngày sản xuất của sản phẩm trong đợt hàng)
* HSD: Date (hạn sử dụng của sản phâm trong đợt hàng)
* SL: int (số lượng sản phẩm nhập về trong đợt hàng)
* isDeleted: bool(dùng để xóa mềm đợt hàng cho biết đợt hàng đã bị xóa hay chưa)
* ptrP: Product \* (con trỏ trỏ tới đối tượng Product mà nó thuộc về)
* List: DList<DetailBill \*> (danh sách các con trỏ trỏ tới các đối tương DetailBill (chi tiết hóa đơn) của những hóa đơn (Bill) đã mua sản phẩm của đợt hàng này

Và các phương thức với

* + Quyền truy cập public:
  + DetailProduct(const String&, const Date&, const Date&, const Date&, const int&)
  + ~DetailProduct()
  + Các cặp hàm Get-Set cho mỗi thuộc tính
  + void Deduct (const int sl) và void Add(const int sl) : giảm và tăng thuộc tính SL một lượng bằng sl
  + virtual bool SLWarn() const : trả về true nếu tống só lượng sản phẩm của các đợt hàng thuộc vê đối tượng Product vả ptrP trỏ tới <=10, ngược lại, trả về fasle
  + bool HSDWarn() const : trả về true nếu còn 7 ngày nữa thì sản phẩm trong đợt hàng sẽ hết hạn sử dụng, ngược lại, trả về false
  + void AddDetailBill(DetailBill \*) : thêm 1 con trỏ tới 1 đối tượng DetailBill vào danh sách List
  + virual int Calculate(const int& sl) const : trả về giá tiền khi mua sản phẩm với số lượng là sl
  + Quyền truy cập private:

void ConnectProduct(Product \*) : tạo liên kết với đối tượng Product mà nó thuộc về

* Lớp DetailWeightProduct

Ngoài các thuộc tính kế thừa từ lớp DetailProduct, lớp này có thêm thuộc tính

* + CalUnit: static int (đơn vị tính giá tiền, được quy ước chung cho mọi sản phẩm là 100 gam)

Lớp này cũng sẽ cung cấp các phương thức tương tự như DetailProduct như override lại cách phương thức

* + virtual bool SLWarn() const : trả về true nếu tống só lượng sản phẩm của các đợt hàng thuộc vê đối tượng Product vả ptrP trỏ tới <=1000 gam, ngược lại, trả về fasle
  + virual int Calculate(const int& sl) const : trả về giá tiền khi mua sản phẩm với khối lượng là sl

#### 4.1.1.4. Lớp Customer

Lớp này gồm các thuộc tính với quyền truy cập pivate:

* + ID: String
  + Name: String
  + Address: String (địa chỉ khách hàng)
  + PhoneNumner: Phone (số điện thoại khách hàng)
  + isDeleted: bool (dùng để xóa mềm khách hàng, cho biết khách hàng này đã bị xóa hay chưa)
  + List : DList<Bill \*> (danh sách các con trỏ trỏ đối tượng Bill, hóa đơn mua hàng, của khách hàng)
  + NumberCustomer: static long long (số khách hàng có trong hệ thống)
  + Và các phương thức với quyền truy cập public:
  + Customer(const String&,const Phone&, const String&);
  + ~Customer()
  + Các cặp hàm Get-Set cho mỗi thuộc tính
  + void AddBill(Bill \*) : thêm 1 con trỏ trỏ tới đối tượng Bill vào List
  + void CancelCurrentBill() : xóa hóa đơn vừa thực hiện (vừa được thêm vào List) của khách hàng

#### 4.1.1.5. Lớp Bill

Lớp này gồm các thuộc tính với quyền truy cập pivate:

* + ID: String
  + Total: int (tổng hóa đơn)
  + BuyDate: Date (Ngày thanh toán hóa đơn)
  + BuyTime: Time (Thời gian lúc thanh toán hóa đơn)
  + ptrC: Customer \* (con trỏ chứa địa chỉ của đối tượng Customer, khách hàng thanh toán hóa đơn)
  + List: DList<DetailBill\*> (danh sách các con trỏ trỏ tới đối tượng DetailBill)
  + NumDetailBill: int (số lượng phần tử trong List – số mặt hàng có trong hóa đơn)
  + NumBill: static long long (số lượng hóa đơn có trong hệ thống)

Và các phương thức với

* Quyền truy cập public:
  + Bill(const Date&, const Time&)
  + ~Bill()
  + Các cặp hàm Get-Set cho mỗi thuộc tính
  + Void AddDetailBill(DetailBill \*) : thêm 1 con trỏ trỏ tới đới tương DetailBill vào List
  + void Cancel() : hủy hóa đơn khi khách hàng không muốn mua hàng
  + void Confirm() : xác nhận việc mua hàng
* Quyền truy cập private:
  + void ConnectCustomer(Customer \*) : tạo liên kết với khách hàng đã thanh toán hóa đơn

#### 4.1.1.6. Lớp DetailBill

Vì trong hóa đơn thường sẽ gồm nhiều mặt hàng mỗi mặt hàng lại được mua với số lượng khác nhau. Vì vậy để dễ dàng trong việc quản lý ta sẽ tạo ra một lớp riêng để quản lý từng mặt hàng trong một hóa đơn, lớp DetailBill.

Lớp này gồm các thuộc tính với quyền truy cập pivate:

* ptrDP: DetailProduct \* (con trỏ trỏ tới đối tượng DetailProduct, chi tiết mặt hàng được mua)
* ptrB: Bill \* (con trỏ trỏ tới hóa đơn mà nó thuộc về)
* SL: int (số lượng sản phẩm được mua của mặt hàng)

Và các phương thức với

* Quyền truy cập public:
  + DetailBill(DetailProduct \*, const int &)
  + ~DetailBill()
  + Các cặp hàm Get-Set cho thuộc tính SL
  + Các hàm trả về thông tin của mặt hàng được trỏ bởi ptrDP
  + int Pay() const : trả về số tiền cần thanh toán (chỉ tính riêng cho mặt hàng trỏ bới ptrDP)
* Quyền truy cập Private:
  + void ConnectBill(Bill \*) : tạo liên kết với hóa đơn mà nó thuộc về

### Đối tượng (Object)

Mục đích chính của phần mềm là quản lý việc kinh doanh của siêu thị – chính là quản lý các đối tượng thuộc các lớp kể trên. Nói cách khác, chương trình sẽ được lập trình để quản lý tập hợp các đối tượng mà cụ thể trong là 3 danh sách liên kết đôi

* DList<Product \*> PList : danh sách chứa các con trỏ. Mỗi trỏ trỏ tới một đối tượng Product có trong hệ thống => quản lý mặt hàng
* DList<Customer \*> CList: danh sách chứa các con trỏ. Mỗi trỏ trỏ tới một đối tượng Customer có trong hệ thống => quản lý khách hàng
* DList<Bill \*> BList: danh sách chứa các con trỏ. Mỗi trỏ trỏ tới một đối tượng Bill có trong hệ thống =>quản lý hóa đơn

Để quản lý các đối tượng trên, chúng cần phải tạo ra tương ứng 3 lớp quản lý mặt hàng, khách hàng và hóa đơn. Tuy nhiên, vì bản chất chúng là các danh sách liên kết đôi, mà chúng ta đã có class template DList, nó có các phương thức để quản lý danh sách liên kết đôi, do đó, chúng sẽ sử dụng luôn lớp này mà không cần tạo ra thêm các lớp quản lý khác.

### Quan hệ (Relation)

Ta có sơ đồ thể hiện quan hệ của các lớp như sau

## Tổ chức chương trình

### Giao diện chính của chương trình

1. Menu bắt đầu

A screenshot of a computer

Description automatically generated

Khi chạy chương trình, menu chính của chương trình sẽ xuất với 5 lựa chọn như trong hình vẽ trên. Lúc này, ta nhập chức năng mà chúng mong muốn và nhấn ENTER để bắt đầu.

1. Menu quản lý mặt hàng

A screenshot of a computer

Description automatically generated

1. Menu quản lý khách hàng

A screenshot of a computer

Description automatically generated

1. Menu quản lý hóa đơn

A screenshot of a computer

Description automatically generated

1. Giao diện tính hóa đơn

A screen shot of a computer

Description automatically generated

### Kết quả thực thi của chương trình

#### Chức năng quản lý sản phẩm (Product Management)

1. Thêm mới sản phẩm (Add new Product(s))

A screen shot of a computer

Description automatically generated

Ở chức năng chúng ta có thể thêm mới từ file (Add from file) hoặc thêm mới bằng cách nhập trực tiếp từ bàn phím (Add from keyboard)

* Add from file

A screen shot of a computer

Description automatically generated

Chương trình sẽ yêu cầu chúng ta nhập tên file lưu dữ liệu cần thêm (file text). Đối với file chứa dữ liệu đầu vào cần có định dạng như hình dưới đây

A screen shot of a computer

Description automatically generated

Tên file đầu vào chính là ngày nhập của đợt hàng (ngày, tháng, năm được phân tách nhau bởi dấu chấm “.”). Hàng đầu tiên của file là định dạng dữ liệu cần phải tuân theo ở mỗi dòng, cụ thể như sau:

* Product ID sẽ gồm 9 kí tự phải được bắt đầu bởi 2 chữ cái “M” và “H” theo sau là 5 chữ số bất kì. Tiếp theo là chữ số “0”. Cuối cùng sẽ là 1 chữ số, “0” hoặc “1”, thể hiện loại mặt hàng – “0” nếu đây là mặt hàng tính theo từng sản phẩm một hay “1” nếu đây là mặt hàng tính theo khối lượng
* Name: tên sản phẩm, được viết không dấu
* Origin: xuất xứ sản phẩm, viết không dấu
* Batch ID sẽ gồm 4 kí tự. Trong đó, kí tự đầu tiên luôn là “L” theo sau bởi 3 chữ số bất kì
* MFG (ngày sản xuất) và EXP (hạn sử dụng) được biểu diễn theo định dạng “ngày/tháng/năm” (dd/mm/yyyy)
* Cuối cùng là giá tiền (Price) và số lượng/khối lượng (Quantity/Weight) của sản phẩm là các số dương

Các thông tin trên được ngăn cách nhau bới dấu phảy “,’.

Sau khi nhập tên file, nếu đúng định dạng file, chương trình sẽ thực hiện và thông báo kết quả

A screen shot of a computer

Description automatically generated

Trong trường hợp dữ liệu trong file đầu vào đã tồn tại hoặc định dạng đầu vào bị sai thì chương trình sẽ đưa ra thông báo tương ứng

A screen shot of a computer

Description automatically generated

* Add from keyboard

Chương trình sẽ yêu cầu chúng nhập lần các thông tin cho sản phẩm cần thêm đầu tiên là nhập ID của sản phẩm. Ở đây nếu chúng ta, để chúng và nhấn ENTER thì chương trình sẽ dừng chức năng này lại và quay lại menu trước đó. Nếu ID chúng ta nhập vào chưa có trong hệ thống thì ta phải lần lượt nhập hết các thông tin, còn nếu mặt hàng đã có trong hệ thống thì chúng ta sẽ chỉ nhập các thông tin chi tiết về sản phẩm trong đợt nhập hàng đó

A black screen with white text

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

1. Chức năng tìm kiếm sản phẩm (Search Product)

A black screen with white text

Description automatically generated

Chức năng này cho phép ta có thể tìm kiếm theo ID, tên và xuất xứ của mặt hàng.

Chúng tôi xin được giới thiệu tìm kiếm theo ID, hai cái còn lại tương tự.

A screen shot of a computer

Description automatically generated

Khi chọn tìm kiếm theo ID, chương trình sẽ yêu cầu chúng ta nhập mã ID cần tìm kiếm. Sau khi tìm kiếm trong danh sách các thông tin liên quan đến mặt hàng sẽ được hiện ra cho chúng ta.

A screenshot of a computer

Description automatically generated

Sau khi, thông tin mặt hàng hiện ra, chương trình cho phép chúng xóa các sản phẩm trong một đợt nhập hàng nào đó hoặc toàn bộ sản phẩm mặt hàng đó. Chúng ta có thể bỏ quả bằng cách chọn “0” và ENTER.

Chúng ta cũng có thể hiển thị tất cả mặt hàng có trong hệ thống bằng cách nhập “\*” khi chương trình yêu cầu chúng ta nhập ID

A screenshot of a computer

Description automatically generated

Đối với kết quả tìm kiếm có nhiều mặt hàng thỏa mãn, chúng ta có thể sắp xếp chúng theo thứ tự tăng hoặc giảm dần của giá tiền

A screenshot of a computer

Description automatically generated

Sau khi sắp xếp theo thứ tự mong muốn chúng ta chọn “0” để tiếp tục. Lúc này chương trình sẽ yêu cầu chúng nhập số thứ tự của mặt hàng trong bảng kết quả tìm kiếm để xem chi tiết hoặc chúng ta có thể chọn “0” để quay trở lại.

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

Sau đó mọi thứ tiếp tục như khi ta tìm kiếm một ID cụ thể.

Ở trên, ta thấy vì sản phẩm trong đợt nhập hàng có Batch ID L001 đã quá hạn dụng nên dòng chứa thông về sản phẩm trong đợt nhập hàng đõ được in đậm với hạn sử dụng được làm nổi bật với màu vàng.

Trong trường hợp số lượng mặt hàng nhỏ hơn hoặc 10 đối với loại sản phẩm tính theo từng sản phẩm và nhỏ hon hoặc bằng 1000 gam (1kg) đối với sản phẩm tính theo khối lượng

A screen shot of a computer

Description automatically generated

1. Quản lý sự hết hạn của sản phẩm (Product expiration management)

A screen shot of a computer

Description automatically generated

Chương trình sẽ yêu cầu chúng ta nhập số ngày, ta kí hiệu là d ngày, và sau đó bảng kết quả gồm các mặt hàng có HSD từ hôm nay cộng d ngày (ví dụ: nếu hôm là 1/1/2024 và d=5 thì ngày mà ta lấy làm mốc là 6/1/2024) sẽ được hiển thị

A screen shot of a computer

Description automatically generated

Lúc này chúng ta có thể chọn “1” để xóa hết tất cả các sản phẩm này khỏi hệ thống hoặc “0” để quay trở lại.

#### Chức năng quản lý khách hàng (Customer Management)

1. Thêm mới khách hàng (Add new Customer(s))

Cũng tương tự như việc thêm mới sản phẩm chúng ta có thể thêm mới từ file hoặc thêm mói trực tiếp từ bàn phím. Đối với file đầu vào, nó có định dạng như sau

A screenshot of a computer

Description automatically generated

Hàng đầu tiên của file là định dạng dữ liệu cần phải tuân theo ở mỗi dòng, cụ thể như sau:

* Name: tên khách hàng, viết không dấu
* Phone: số điện thoại khách hàng, phải gồm 10 chữ số
* Address: địa chỉ khách hàng

Các thông tin được ngăn cách nhau bởi dấu phẩy “,”

Vì cách thực hiện tương tự nên chúng sẽ không giới thiệu chi tiết mà chỉ sử dụng hình minh họa.

* Add from file

A screen shot of a computer

Description automatically generated

* Add from keyboard

A screen shot of a computer

Description automatically generated

1. Tìm kiếm khách hàng (Search Customer)

Ta có thể tìm kiếm khách hàng theo số điện thoại và tên

* Theo số điện thoại

Để tìm kiếm khách hàng mong muốn ta nhập số điện thoại cần tìm và nhấn ENTER

A screenshot of a computer

Description automatically generated

Lúc này, nếu khách hàng với số điện thoại đó có trong hệ thống thì thông tin của khách hàng cùng lịch sử mua hàng của khách hàng đó sẽ được hiển thị. Sau khi hiển thị, chương trình sẽ hỏi chúng ta có muốn xóa khách hàng này hay không. Nếu có thì chúng ta nhấn phím “d” để thực hiện.

Cũng giống như tìm kiếm mặt hàng, chúng ta có thể hiển thị toàn khách hàng trong hệ thống bằng cách nhập dấu “\*” khi chương trình yêu cầu nhập số điện thoại cần tìm kiếm.

A screenshot of a computer

Description automatically generated

Sau đó, muốn xem chi tiết khách hàng nào chúng ta chỉ cần nhập số thứ tự của khách hàng trên bẳng kết quả và nhấn ENTER

A screenshot of a computer

Description automatically generated

* Theo tên
* A screen shot of a computer

  Description automatically generated

1. Cập nhật thông tin khách hàng (Update information of Customer)

A screen shot of a computer

Description automatically generated

Đầu tiên, chúng ta cần phải số điện thoại của khách hàng cần cập nhật thông tin.

A screen shot of a computer

Description automatically generated

Các thông tin liên quan đến khách hàng này sẽ được hiển thị và chúng ta chọn mục thông tin nào cần cập nhật

A screen shot of a computer

Description automatically generated

Sau khi chọn xong, chúng ta chỉ cần nhập thông tin và nhấn ENTER để lưu. Trong trường nếu chúng không muốn sửa đổi thông tin nữa thì chỉ cần để trống và nhấn ENTER.

#### Chức năng quản lý hóa đơn (Bill Management)

1. Tính doanh thu (Calculate Revenue)

Để tính doanh thu chúng ta chỉ cần nhập lần lượt ngày bắt đầu và kết thúc của khoảng thời gian muốn tính doanh thu. Lúc này, chương sẽ tìm kiếm các hóa đơn trong khoảng thời gian đó tính toán và đưa ra tổng doanh thu cùng với thông tin một cách tổng quan về các hóa đơn trong khoảng thời gian đó.

A screenshot of a computer

Description automatically generated

1. Tìm kiếm hóa đơn (Search Bill)

Ta có thể tìm kiếm theo ID, số điện thoại của khách hàng và ngày mua hàng

* Theo ID

Nhập ID của hóa đơn cần tìm. Chương trình sẽ tìm kiếm hóa đơn và hiển thị lên màn hình các thông tin chi tiết của nó.

A screenshot of a computer

Description automatically generated

* Theo số điện thoại khách hàng

Nhập số điện thoại khách hàng cần tìm kiếm hóa đơn. Sau đó danh sách các hóa đơn của khách hàng đó sẽ được hiển thị.

A screenshot of a computer

Description automatically generated

Khi này, chúng ta nhập số thứ tự của hóa đơn trong bảng kết quả để xem chi tiết

A screenshot of a computer

Description automatically generated

* Theo ngày mua

Các bước thực hiện cũng tương tự như tìm kiếm theo số điên thoại khách hàng

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Ngoài ra, sau khi tìm kiếm hóa đơn, chương trình hỏi chúng ta có muốn xuất hóa đơn hay không. Nếu có thì ta chỉ cần nhấn phím “p” để thự hiện. Hóa đơn sẽ được xuất ra một file có tên là ID của hóa đơn

A screen shot of a computer

Description automatically generated

#### Chức năng thanh toán (Payment)

Để thanh toán, đầu tiên, chúng ta cần phải nhập số điện thoại của khách hàng muốn thanh toán. Nếu số điện thoại khách hàng đã có trong hệ thống thì các thông tin của khách hàng sẽ được tự đông hiển thị. Ngược lại, chúng ta phải nhập các thông tin như tên, địa chỉ của khách hàng và khách hàng nàu sẽ được thêm vào hệ thống. Trong trường hợp khách hàng không muốn lưu thông tin thì ta chỉ cần để trống khi chương trình yêu cầu nhập số điện thoại và nhấn ENTER. Nếu là khách hàng mới và họ chỉ cung cấp số điện thoại mà không cung cấp tên thì khách hàng này cũng sẽ không được lưu vào hệ thống

A screen shot of a computer

Description automatically generated

A screen shot of a black screen

Description automatically generated

Tiếp theo, chúng ta phải nhập ID của sản phẩm. ID sản phẩm này hơi khác một chút so với ID của mặt hàng như đã đề cập ở những phần trên. Ở đây vì sản phẩm mà khách hàng mua sẽ thuộc 1 lô hàng (đợt nhập hàng) của 1 mặt hàng nào đó. Do đó, ở đây ID của sản phẩm sẽ gồm cả Product ID nối với Batch ID (ví dụ sản phẩm có Product ID là MH0000100 và sản phẩm đó được nhập trong đợt hàng (lô hàng) có Batch ID là L001 thì ID của sản phẩm đó sẽ là MH0000100L001). Vậy ID sản phẩm sẽ gồm 13 kí tự (9 kí tự của Product ID và 4 kí tự của Batch ID) và sẽ có dạng “MH\*\*\*\*\*0\*L\*\*\* ”.

A screenshot of a computer

Description automatically generated

Sau khi nhập ID sản phẩm xong, thông tin của nó sẽ được hiển thị và chúng ta phải nhập số lượng(đối sản phẩm tính theo từng sản phẩm) hoặc khối lượng (đối với sản phẩm tính theo khối lượng) để thanh toán. Ta thực hiện lập lại như vậy cho đến khi nhập hết mặt hàng muốn mua. Lúc này, ta chỉ cần để trống ở lần yêu cầu nhập ID sản phẩm tiếp theo và nhấn ENTER. Hóa đơn sẽ được hiển thị.

A screenshot of a computer

Description automatically generated

Chương trình sẽ hỏi ta có muốn xác nhận mua hàng hay không. Nếu có thì ta nhấn phím “y”. Trong trường hợp, nếu chúng ta muốn hủy hóa đơn thì chỉ cần nhấn 1 phím bất kì khác “y”. Sau khi xác nhận, hóa đơn sẽ được lưu vào hệ thống.

A screenshot of a computer

Description automatically generated

Chương trình sẽ hỏi ta có muốn xuất hóa đơn hay không. Tương tự như chúng tôi đã nhắc đến ở trên nếu có thì chúng ta nhấn phím “p”. Ngược lại, nhấn một phím bất kì khác “p”.

#### Kết thúc chương trình (Quit)

Trước khi chương trình kết thúc, danh sách các mạt hàng cũng như khách hàng có trong hệ thống sẽ dược xuất ra 2 file “OuputCustomer.txt” và “OutputProduct.txt”

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### Nhận xét

Chương trình thực hiện cho ra kết quả chính xác. Ngoài ra, chương trình “thân thiện” và dễ sử dụng đối với người dung. Tuy nhiên, giao diện vẫn còn đơn giản, chưa bắt mắt

# 5.  KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết luận

Sau một khoảng thời gian dài phát triển và thử nghiệm thực tế, chương trình đã đáp ứng hầu hết nhu cầu của đối tượng. Trong quá trình nghiên cứu và thực hiện đồ án, chúng em đã tìm hiểu và nắm vững cấu trúc dữ liệu danh sách liên kết đôi, lập trình hướng đối tượng và các kiến ​​thức liên quan khác... Cuối cùng, chúng em đã hoàn thành chương trình: “*Hệ thống quản lý siêu thị (Minimart Management System)”*. Tuy đã chương trình đã hoàn thiện nhưng vẫn còn nhiều giải pháp tối ưu hơn về mặt thuật toán lẫn về giao diện nên cần được tiếp tục phát triển trong tương lai.

## 5.2. Hướng phát triển

* Tiếp thu ý kiến, phát triển thêm các tính năng mới.
* Phát triển và đổi mới giao diện.
* Tiếp tục phát triển thuật toán nhằm tối ưu thời gian xử lí chương trình.

**TÀI LIỆU THAM KHẢO**

[1] Lê Minh Hoàng, *Giải thuật & Lập trình*, Đại học Sư phạm Hà Nội, 2003.

[2] Nguyễn Thanh Bình, Phạm Minh Tuấn, Đặng Thiên Bình, *Phân tích và thiết  
kế giải thuật*, NXB GD, 2016.

[3]Lê Thị Mỹ Hạnh, Giáo trình môn học Lập trình hướng đối tượng, 2002.