Bộ Giáo Dục Và Đào Tạo

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN

Khoa Công Nghệ Thông Tin

**- - - 🙞 🕮** **🙜 - - -**

Logo

Description automatically generated

BÁO CÁO ĐỒ ÁN TIN HỌC

QUẢN LÝ CỦA HÀNG XE MÁY HONDA

|  |  |  |  |
| --- | --- | --- | --- |
| **Giảng Viên Hướng Dẫn :** | Hoàng Khuê | | |
| **Thành Viên Nhóm 1** : | | **Họ Và Tên** | **MSSV** | |
|  | | Nguyễn Hòa Ninh Đan | DH52007049 | |
|  | |
|  | |
|  | |
|  | |

*Năm 2022-2023*

Bộ Giáo Dục Và Đào Tạo

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN

Khoa Công Nghệ Thông Tin

**- - - 🙞 🕮** **🙜 - - -**

Logo

Description automatically generated

BÁO CÁO ĐỒ ÁN TIN HỌC

QUẢN LÝ CỦA HÀNG XE MÁY HONDA

|  |  |  |  |
| --- | --- | --- | --- |
| **Giảng Viên Hướng Dẫn :** | Hoàng Khuê | | |
| **Thành Viên Nhóm 1** : | | **Họ Và Tên** | **MSSV** | |
|  | | Nguyễn Hòa Ninh Đan | DH52007049 | |
|  | |
|  | |
|  | |
|  | |

*Năm 2022-2023*

LỜI CAM ĐOAN

Tôi xin cam đoan đồ án tin học : Đề tài “Quản lý cửa hàng xe máy Honda” được tiến hành công khai , dựa trên sự cố gắng , nỗ lực không ngừng của đội nhóm và sự giúp đỡ không nhỏ từ phía trường Đại học Công Nghệ Sài Gòn, khoa công nghệ thông tin nói chung và dưới sự hướng dẫn nhiệt tình của giảng viên Hoàng Khuê .

Các sô liệu và kết quả nghiên cứu của đề tài là trung thực và hoàn toàn không sao chép hoặc sử dụng kết quả của đề tài nghiên cứu nào tương tự. Ngoài ra trong trong báo cáo có sử dụng một số tài liệu tham khảo được trích dẫn nguồn và chú thích rõ ràng .Nhóm xin hoàn toàn chịu trách nhiệm trước bộ môn khoa và nhà trường về sự cam đoan này.

|  |
| --- |
| *Hồ Chí Minh , Ngày 15 Tháng 10 Năm 2022* |
| Học viên thực hiện |
| **Nguyễn Hòa Ninh Đan** |

Lời Cảm Ơn

Trên thực tế không có sự thành công nào mà không gắn liền với sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô, gia đình và bạn bè.

Với lòng biết ơn sâu sắc nhất, chúng em xin gửi đến quý Thầy Cô ở Khoa Công Nghệ Thông Tin – Trường Đại Học Công Nghệ Sài Gòn đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức tổ chức cho chúng em được tiếp cận với môn học mà theo nhóm em là hữu ích đối với sinh viên ngành Công Nghệ Thông Tin. Đó là môn học “Đồ án tin học” .

Nhóm xin chân thành cảm ơn giảng viên Hoàng Khuê đã tận tâm hướng dẫn chúng em qua từng buổi trao đổi, thảo luận về đồ án tin học. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì chúng em nghĩ bài báo cáo này rất khó có thể hoàn thiện được. Một lần nữa, nhóm xin chân thành cảm ơn thầy.

Bài thu hoạch được thực hiện trong khoảng thời gian 10 tuần. Bước đi vào thực tế, tìm hiểu đồ án tin học, kiến thức của cả nhóm còn hạn chế. Do vậy, không tránh khỏi những thiếu sót là điều chắc chắn, nhóm rất mong nhận được ý kiến đóng góp quý báu của thầy cô và các bạn để kiến thức của nhóm trong đồ án lần này được hoàn thiện hơn.

Sau cùng , chúng em xin kính chúc quý thầy cô trong Khoa Công Nghệ Thông Tin và thầy Hoàng Khuê thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Chân thành cảm ơn!

Muc lục

[Chương I. TỔNG QUAN 1](#_Toc123661464)

[1. Đề Tài 1](#_Toc123661465)

[1.1. Lý do chọn đề tài 1](#_Toc123661466)

[1.2. Khảo Sát Thực Tiễn 1](#_Toc123661467)

[1.2.1. Giới thiệu về cửa hàng 1](#_Toc123661468)

[1.2.2. Quy trình của cửa hàng qua khảo sát 1](#_Toc123661469)

[1.2.3. Ưu nhược điểm của hệ thống cũ 2](#_Toc123661470)

[1.2.3.1. Ưu điểm 2](#_Toc123661471)

[1.2.3.2. Nhược điểm 2](#_Toc123661472)

[1.2.4. Một số mẫu hóa đơn của hệ thống cũ 2](#_Toc123661473)

[1.2.5. Đề xuất giải pháp doanh nghiệp 3](#_Toc123661474)

[1.2.6. Phạm vi đề tài 3](#_Toc123661475)

[2. Môi Trường Phát Triển Ứng Dụng 3](#_Toc123661476)

[2.1. IDE 3](#_Toc123661477)

[3. Ngôn ngữ và giao diện 4](#_Toc123661478)

[Chương II. CƠ SỞ LÝ THUYẾT 4](#_Toc123661479)

[1. Cấu trúc lưu trữ 4](#_Toc123661480)

[1.1. Danh sách liên kết 4](#_Toc123661481)

[1.1.1. Lý thuyết 4](#_Toc123661482)

[ Tính cấp phát dữ liệu động 4](#_Toc123661483)

[Tính liên kết của phần tử đầu và phần tử đứng sau 5](#_Toc123661484)

[1.1.2. Ưu điểm và nhược điểm của danh sách liên kết 5](#_Toc123661485)

[1.2. Cây 5](#_Toc123661486)

[1.2.1. Lý thuyết 5](#_Toc123661487)

[1.3. Cây nhị phân 6](#_Toc123661488)

[1.3.1. Lý thuyết 6](#_Toc123661489)

[1.4. Cây nhị phân tìm kiếm 7](#_Toc123661490)

[1.4.1. Lý thuyết 7](#_Toc123661491)

[1.4.2. Ưu điểm và nhược điểm của cấu trúc dữ liệu cây 7](#_Toc123661492)

[1.5. Cây đỏ đen –(tìm hiểu thêm)- 8](#_Toc123661493)

[1.5.1. Lý thuyết 8](#_Toc123661494)

[1.5.2. Ưu điểm và nhược điểm của cây đỏ đen 9](#_Toc123661495)

[1.5.3. Ứng dụng của cây đỏ đen 9](#_Toc123661496)

[2. Thuật toán 9](#_Toc123661497)

[2.1. Duyệt trung tự trên cây nhị phân (LNR) 9](#_Toc123661498)

[2.1.1. Lý thuyết 9](#_Toc123661499)

[2.1.2. Ưu điểm và nhược điểm của duyệt cây 10](#_Toc123661500)

[2.2. Tìm kiếm node trên cây nhị phân tìm kiếm 10](#_Toc123661501)

[2.2.1. Lý thuyết 10](#_Toc123661502)

[2.2.2. Ưu điêm và nhược điểm của thuật toán tìm kiếm 11](#_Toc123661503)

[2.3. Sắp xếp trộn – MergeSort -(tìm hiểu thêm)- 12](#_Toc123661504)

[2.3.1. Lý thuyết 12](#_Toc123661505)

[2.3.2. Ưu điểm và nhược điểm thuật toán sắp xếp trộn 13](#_Toc123661507)

[Chương III. TRÌNH BÀY HỆ THỐNG VÀ GIAO DIỆN HỆ THỐNG 14](#_Toc123661508)

[1. Thiết kế 14](#_Toc123661509)

[1.1. Menu đa cấp 14](#_Toc123661510)

[1.2. Class Diagram 15](#_Toc123661511)

[2. Giao diện App 16](#_Toc123661512)

[Chương IV. CÀI ĐẶT THỬ NGHIỆM 18](#_Toc123661513)

[1. Cài đặt 18](#_Toc123661514)

[1.1. Chia cấu trúc file 18](#_Toc123661515)

[1.2. Tiến hành cài đặt code 19](#_Toc123661516)

[2. Thử nghiệm 38](#_Toc123661517)

[Chương V. TỰ ĐÁNH GIÁ 38](#_Toc123661518)

[1. Những kết quả đạt được 38](#_Toc123661519)

[2. Những mặt hạn chế 38](#_Toc123661520)

[3. Hướng phát triển tiếp theo 38](#_Toc123661521)

# TỔNG QUAN

## Đề Tài

## Lý do chọn đề tài

Ngành công nghệ thông tin đã giúp cho nhiều tổ chức, doanh nghiệp tăng lợi nhuận, mở rộng kinh doanh, giảm bớt chi phí về nhân lực và xử lý hiệu quả các công việc phức tạp nhanh chóng…Tin học ngày càng được ứng dụng phổ biến trong cuộc sống, không chỉ trong doanh nghiệp lớn mà có cả các doanh nghiệp vừa và nhỏ, các cửa hàng…

Như ta đã biết vấn đề đi lại, cũng như phương tiện vận chuyển là nhu cầu có từ xa xưa mà mỗi hình thức ở các thời đại khác nhau, ở thời điểm hiện tại các phương tiện đi lại vô cùng đa dạng như đặc biệt với một nước đang phát triển như Việt Nam ta thì xe máy là phương tiện của đại đa số dân ta sử dụng để đi lại, với giá thành khá rẻ cũng như dễ dàng sử dụng …. Các cửa hàng xe máy dần mọc lên. Trong thực tế, việc quản lý cửa hàng với khối lượng công việc lớn mà vẫn phải thực hiện theo phương pháp ghi chép thủ công, vừa tốn thời gian, công sức lại hay xảy ra nhầm lẫn, sai sót. Vì vậy mà nhóm chúng em tạo ra một chương trình cho cửa hàng xe máy honda, tiết kiệm thời gian công sức, các việc dễ dàng chuyên nghiệp và hiệu quả hơn. Điều này phù hợp hơn với yêu cầu của người bán mặt hàng này.

## Khảo Sát Thực Tiễn

## Giới thiệu về cửa hàng

Đây là một cửa hàng xe máy đại lý của hãng xe Honda thành lập vào những năm 2009, đến nay mở rộng quy mô kinh doanh xe máy nhiều hơn mẫu mã phong phú hơn. Đồng thời lượng khách tin dùng cũng tăng đáng kể. Ông chủ ở đây nhận thấy tiềm năng của tin học hóa vào cửa hàng là cần thiết, nên ông muốn có một hệ thống quản lý quá trình bán xe cũng như thanh toán cho khách hàng. Bởi trước đây mỗi lần bán xe nhân viên cửa hàng phải ghi chép vào sổ theo dõi việc bán hàng. Vì vậy nếu không theo dõi cẩn thận sẽ xảy ra sai sót đáng tiếc bởi số lượng dữ liệu của cửa hàng khá lớn.

## Quy trình của cửa hàng qua khảo sát

1. Khách hàng chọn xe đã trưng bày trong sảnh
2. Khách hàng đến quầy thanh toán tiền.
3. Nhân viên viết đơn thanh toán .
4. Sau khi khách hàng đã thanh toán thành công thì nhân viên xác nhân bằng cách kí vào hóa đơn.
   1. Một bản của hóa đơn được cửa hàng giữ lại, một bản khách hàng giữ.
5. Sau khi khách hàng đã nhận xe . Nhân viên ghi lại lịch sử giao dịch vào sổ.
   1. Nếu khách hàng đó mua 100 triệu thì được chép vào danh sách khách hàng thân thiết của cửa hàng.

## Ưu nhược điểm của hệ thống cũ

## Ưu điểm

* Dễ dàng triển khai .
* Chí phí thấp .
* Thích hợp cho các cửa hàng nhỏ .

## Nhược điểm

* Khi mà quy mô cửa hàng tăng, hệ thông cũ không thể đáp ứng được các yêu cầu khổng lồ, nhanh chóng .
* Sai sót, làm thất thoát.
* Chi phí nhân lực cao.
* Thao tác công việc nhiều.
* Khó khăn cho nhà quản lý

## Một số mẫu hóa đơn của hệ thống cũ

|  |
| --- |
| **CỬA HÀNG STU**  **Đại Lý Phân Phối Xe Máy Honda Chính Hãng**  Địa chỉ Quận 8 ,TPHCM  Số điện thoại 012345689  **HÓA ĐƠN THANH TOÁN** |
| |  |  |  |  | | --- | --- | --- | --- | | Mã khách hàng:  Tên khách hàng:  Địa chỉ :  Số điện thoại: |  | Mã Nhân Viên:  Tên Nhân Viên:  Số điện thoại: |  | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Mã xe | Tên xe | Số lượng | Đơn giá | Tổng tiền | |  |  |  |  |  | |
| |  |  | | --- | --- | | Khách hàng kí tên | ….,ngày …tháng…năm  Người lập hóa đơn | |

## Đề xuất giải pháp doanh nghiệp

Từ những nhược điểm phát sinh trong quá trình làm truyền thống như vậy, nhóm chúng tôi đưa ra đề xuất giải quyết các vấn đề ấy cho cửa hàng :

Hệ thống cung cấp chức năng nhân viên bán xe máy:

* Quản lý hóa đơn cho mỗi phiên giao dịch
* Tính tiền hóa đơn
* Quản lý khách hàng vip, hóa đơn mua xe trên 100 triệu

## Phạm vi đề tài

Thông qua những đề xuất ở trên mà chúng tôi sẽ tự động hóa một số phần sau :

Hệ thống thanh toán bao gồm :

* Hóa đơn thanh toán
  + Chức năng gồm :tìm, thêm, sửa, xóa.
  + Thông tin hóa đơn, thông tin khách hàng, thông tin xe máy và thành tiền hóa đơn.
  + Đầu vào thông tin hóa đơn,
  + Đầu ra danh sách hóa đơn và thành tiền trên mỗi hóa đơn.
* Danh sách VIP
* Xem danh sách khách hàng ra màn hình.
* Thông tin khách hàng gồm :Họ tên khách hàng, số điện thoại.
* Đầu vào là thông tin hóa đơn được tóm gọn,
* Đầu ra là danh sách hóa đơn VIP.

## Môi Trường Phát Triển Ứng Dụng

## IDE

* Ở đây chúng tôi sử dụng Visual studio 2022 vì một số lý do :
* Đây là IDE các các lập trình viên sử dụng nhiều nhất thế giới.
* Chủ yếu sử dụng để lập trình ngôn ngữ C++, C#.
* Hỗ trợ nhiều ngôn ngữ lập trình.
* Hỗ trợ trên đa nền tảng .
* Giao diện thân thiện.
* Visual studio 2022 là phiên bản mới của nhà microsoft được hỗ trợ trên các hệ điều hành x64-bit:
* Windows 11 bản 21H2 trở lên.
* Windows 10 bản 1909 trở lên.
* Windows Server 2022-2019-2016.
* Phần cứng yêu cầu
* Bộ xử lý x64 1,8GHZ trở lên (khuyên lõi tứ hoặc tốt hơn).
* Ram tối thiểu 4GB, khuyên dùng 16GB.
* Dung lượng đĩa cứng tối thiểu là 850MB tối đa 210GB.
* Dung lượng trống yêu cầu 20-50GB.

## Ngôn ngữ và giao diện

* Ngôn ngữ sử dụng trong quá trình thực hiện hóa chương trình là C++ .
* Chúng tôi chọn C++ vì nó là ngôn ngữ thuộc lập trình hướng đối tượng, từ kỹ thuật OOP của ngôn ngữ C++ cho phép tạo ra các tối tượng trong code trừu tượng hóa đối tượng thực tế cuộc sống. Mà bài toán đặc ra của cửa hàng xe máy cũng thuộc một vấn đề trong cuộc sống.
* Về phần giao diện bởi thực hiện trên C++ nên giao diện console là tiên quyết tuy không hoàn hảo nhưng khá hiệu quả.

# CƠ SỞ LÝ THUYẾT

## Cấu trúc lưu trữ

## Danh sách liên kết

## Lý thuyết

* **Khái niệm**
* Mỗi nút (phần tử) trong danh sách liên kết có tối thiểu hai vùng. Vùng thứ nhất chứa dữ liệu và vùng thứ hai là vùng liên kết dùng để kết nối các nút với nhau.
* Khi liên kết trỏ đến phần tử kế tiếp mà NULL thì nghĩa là đó là phần tử cuối cùng của danh sách liên kết.
* **Đặc điểm**

### Tính cấp phát dữ liệu động

* Được cấp phát bộ nhớ khi chạy chương trình
* Có thể thay đổi kích thước qua việc thêm, xóa phần tử
* Kích thước tối đa phụ thuộc vào bộ nhớ khả dụng của RAM
* Các phần tử được lưu trữ ngẫu nhiên (không liên tiếp) trong RAM

### Tính liên kết của phần tử đầu và phần tử đứng sau

* Chỉ cần nắm được phần tử đầu và cuối là có thể quản lý được danh sách
* Truy cập tới phần tử ngẫu nhiên phải duyệt từ đầu đến vị trí đó
* Chỉ có thể tìm kiếm tuyến tính một phần tử

Diagram, box and whisker chart

Description automatically generated

*Hình minh họa danh sách liên kết*

## Ưu điểm và nhược điểm của danh sách liên kết

* **Ưu điểm**
* Cung cấp giải pháp để chứa cấu trúc dữ liệu tuyến tính.
* Dễ dàng thêm hoặc xóa các phần tử trong danh sách mà không cần phải cấp phát hoặc tổ chức lại trật tự của mảng.
* Cấp phát bộ nhớ động
* **Nhược điểm**
* Một danh sách liên kết đơn giản không cho phép truy cập ngẫu nhiên dữ liệu.
* Chính vì lí do trên mà một số phép tính như tìm phần tử cuối cùng, xóa phần tử ngẫu nhiên hay chèn thêm, tìm kiếm có thể phải duyệt tất cả các phần tử.
* Do những ưu nêu trên mà nhóm tôi đã chọn danh sách liên kết để áp dụng cho việc lưu danh sách khách vip.

## Cây

## Lý thuyết

* **Khái niệm**

Cấu trúc cây (Tree) là một tập hợp các phần tử gọi là nút (node), mỗi cây có một nút gốc (root) chứa nhiều nút con, mỗi nút con lại là một tập hợp các nút khác gọi là cây con (subtree).

* Bậc của nút: là số nút con của nút đó.
* Bậc của cây: là bậc lớn nhất của nút trong cây đó, cây bậc n sẽ được gọi là cây n – phân.
* Nút lá: nút lá là nút có bậc bằng 0.
* Nút nhánh: là nút có bậc khác 0 mà không phải nút gốc (hay còn gọi là nút trung gian).
* Mức của nút: là số nguyên đếm từ 0, các nút ngang hàng nhau thì có cùng mức.
* Chiều cao (chiều sâu): là mức lớn nhất của các nút lá.
* Độ dài đường đi đến nút x: là số nhánh (cạnh nối hai nút) cần đi qua tính từ nút gốc đến nút x. Hay độ dài đường đi đến nút mức i chính là i.

Diagram

Description automatically generated

*Hình minh họa cấu trúc cây*

## Cây nhị phân

## Lý thuyết

* **Khái niệm**

Cây nhị phân là một trường hợp đặc biệt của cấu trúc cây và nó cũng phổ biến nhất. Đúng như tên gọi của nó, cây nhị phân có bậc là 2 và mỗi nút trong cây nhị phân đều có bậc không quá 2.

Shape

Description automatically generated with medium confidence

*Hình minh họa cây nhị phân*

## Cây nhị phân tìm kiếm

## Lý thuyết

* **Khái niệm**

Cây nhị phân tìm kiếm là cây nhị phân mà trong đó, các phần tử của cây con bên trái đều nhỏ hơn phần tử hiện hành và các phần tử của cây con bên phải đều lớn hơn phần tử hiện hành. Do tính chất này, cây nhị phân tìm kiếm không được có phần tử cùng giá trị.

* **Đặc điểm**
* Tạo nút giống với cây nhị phân
* Nút trái cùng là nhỏ nhất và nút phải cùn là lớn nhất
* Duyệt cây LNR để danh sách có thứ tự

Diagram

Description automatically generated

*Hình minh họa cây nhị phân tìm kiếm*

## Ưu điểm và nhược điểm của cấu trúc dữ liệu cây

* **Ưu điểm**
* Thao tác tìm kiếm, hủy, chèn rất nhanh, có thể cấp phát thêm nút hoặc thu hồi bộ nhớ đã cấp phát cho nút và kích thước cây không bị giới hạn.
* **Nhược điểm**
* Tốn thêm bộ nhớ để lưu nút con trái, nút con phải.

🡪 Bằng ấy những ưu điểm đó nên nhóm hội ý quà quyết định lấy cấu trúc cây để thao tác lưu trữ và sử lý hóa đơn.

## Cây đỏ đen –(tìm hiểu thêm)-

## Lý thuyết

* **Khái niệm**
* Cây đỏ đen là một cây nhị phân tìm kiếm nhưng tuân thủ theo các quy tắc sau:

1. Mọi Node trong cây phải là đỏ hoặc đen.
2. Node gốc là Node đen.
3. Các Node ngoài phải (NULL Node) luôn luôn đen.
4. Nếu một Node là đỏ thì những Node con của nó phải là đen (quy tắc xung đột).
5. Mọi đường dẫn từ Node gốc đến Node ngoài phải có cùng số lượng Node đen.

Diagram

Description automatically generated

*Hình minh họa cây đỏ đen*

**Một số khái nhiệm khác:**

* Chiều cao đen là số Node đen trên đường đi từ x đến Node ngoài
* Hiện tượng xung đột đỏ - đỏ là trường hợp vi phạm quy tắc số 4, khi Node cha và Node con trực tiếp cùng một màu đỏ.

**Một số thao tác trên cây đỏ đen**

* Tìm kiếm và duyệt cây (giống BST).
* Thêm Node mới (insert Node).
* Xóa Node (delete Node).

Còn thao tác thêm Node và xóa Node thì khác so với cây nhị phân tìm kiếm, vì sau mỗi lần thêm Node hoặc xóa Node ta phải cập nhật lại thuộc tính color để không vi phạm các quy tắc của cây đỏ đen.

## Ưu điểm và nhược điểm của cây đỏ đen

* **Ưu điểm**
* Chi phí tìm kiếm O(log2N)
* Insert O(log2N)
* Delete O(log2N)
* Minimum O(log2N)
* Maximum O(log2N)
* **Nhược điểm**
* Phải lưu trữ thuộc tính màu (color) và con trỏ đến nút cha (pParent)
* Cài đặt phức tạp hơn cây AVL và BTS

## Ứng dụng của cây đỏ đen

* Để triển khai bản đồ hữu hạn.
* Để triển khai các gói package trong Java bao gồm java.util.TreeMap và java.util.TreeSet.
* Để triển khai thư viện mẫu chuẩn (STL) trong C++ bao gồm multiset, map, multimap.
* Sử dụng trong nhân Linux.

## Thuật toán

## Duyệt trung tự trên cây nhị phân (LNR)

## Lý thuyết

* **Khái niệm**

Thuật toán duyệt cây nhị phân có 3 loại, duyệt tiền tự (NLR), duyệt hậu tự (LRN) và duyệt trung tự (LNR), ở đây nhóm chỉ giới thiệu về duyệt cây nhị phân theo trung tự (LNR).

**Duyệt trung tự (LNR):**

* Duyệt trung tự cây con trái, duyệt nút gốc, duyệt trung tự cây con phải.

**+Giải thuật**

1. Duyệt cây con bên trái r theo trung thứ tự (nếu có)
2. Thăm góc r
3. Duyệt cây con bên phải r theo trung thứ tự (nếu có)

**Ví dụ**

Chart, radar chart

Description automatically generated

* Thứ tự sau khi duyệt LNR kết thúc là A, B, D, E,G, C, F, H, I.

## Ưu điểm và nhược điểm của duyệt cây

* **Ưu điểm**
* Duyệt nhanh
* Chính xác
* Sắp xếp tăng dần khi duyệt cây, không cần phải ứng dụng thuật toán sắp xếp phức tạp.
* Vì những ưu điểm nêu trên mà nhóm lựa chọn để duyệt và sắp xếp đối tượng khi in ra màn hình.

## Tìm kiếm node trên cây nhị phân tìm kiếm

## Lý thuyết

* **Khái niệm**

Để tìm một phần tử trong cây nhị phân tìm kiếm, chúng ta sẽ thực hiện tương tự việc tìm kiếm nhị phân. Nếu như nút cần tìm nhỏ hơn nút đang xét, chúng ta sẽ tìm cây con bên trái, ngược lại chúng ta sẽ tìm trong cây con bên phải, nếu đúng nút cần tìm thì mình sẽ trả về địa chỉ của nút đó.

**+Giải thuật**

1. Nếu Node hiện tại có giá trị = giá trị cần tìm, trả về true và kết thúc.
2. Nếu Node hiện tại có giá trị > giá trị cần tìm, gọi đệ quy tìm ở cây con bên trái.
3. Nếu Node hiện tại có giá trị < giá trị cần tìm, gọi đệ quy tìm ở cây con bên phải

Nếu tìm đến hết cây(Node đó = NULL) mà không xảy ra (1), trả về false và kết thúc

Chart

Description automatically generated

*Hình minh họa thuật toán tìm kiếm nút 4 trên cây nhị phân tìm kiếm*

## Ưu điêm và nhược điểm của thuật toán tìm kiếm

* **Ưu điểm**
* Cây tìm kiếm nhị phân như cái tên, sinh ra để thực hiện tìm kiếm và cập nhật dữ liệu nhanh chóng.
* Với ưu điểm này nên nhóm quyết định chọn thuật toán tìm kiếm trên cây nhị phân tìm kiếm để, tìm kiếm một phần tử dễ dàng nhanh hơn .

## Sắp xếp trộn – MergeSort -(tìm hiểu thêm)-

## Lý thuyết

**Khái niệm**

* Merge Sort là một thuật toán Chia và Chinh phục. Nó chia mảng đầu vào thành hai nửa, gọi chính nó cho hai nửa và sau đó hợp nhất hai nửa đã sắp xếp. Hàm merge () được sử dụng để hợp nhất hai nửa. Hợp nhất (arr, l, m, r) là quá trình quan trọng giả định rằng arr [l..m] và arr [m + 1..r] được sắp xếp và hợp nhất hai mảng con đã sắp xếp thành một.

**+Giải thuật**

MergeSort (arr [], l, r)

Nếu r> l

1. Tìm điểm giữa để chia mảng thành hai nửa:

Ở giữa m = (l + r) / 2

2. Hợp nhất cuộc gọi Sắp xếp cho nửa đầu:

Gọi mergeSort(arr, l, m)

3. Hợp nhất cuộc gọi Sắp xếp cho nửa sau:

Gọi mergeSort(arr, m + 1, r)

4. Hợp nhất hai nửa được sắp xếp ở bước 2 và 3:

Gọi merge(arr, l, m, r)

**Ví dụ một dãy số {38, 27, 43, 3, 9, 82, 10}**

## Diagram, schematic Description automatically generated

*Hình minh họa thuật toán sắp xếp trộn*

## Ưu điểm và nhược điểm thuật toán sắp xếp trộn

* **Ưu điểm**
* Độ phức tạp trung bình O(n log n), tốc độ giải quyết khá nhanh
* Có tính ổn định và thích ứng, tốc độ không bị ảnh hưởng nhiều bởi dữ liệu đầu vào
* Xử lý khá tốt với dữ liệu lớn đặc biệt là dạng list, file
* **Nhược điểm**
* Tốn nhiều bộ nhớ nếu sử dụng đệ quy
* Code khó cài đặt, tương đối phức tạp

# TRÌNH BÀY HỆ THỐNG VÀ GIAO DIỆN HỆ THỐNG

## Thiết kế

## Menu đa cấp

Diagram

Description automatically generated

* **Ý tưởng mennu đa cấp**
* Menu dự định sẽ có 4 cấp độ
* Cấp thứ 1 là Menu Main.
  + Trong MenuMain có:
    - Cấp thứ 2 là Menu Vip và Menu Pay.
      * Trong Menu Pay có :
        + Cấp thứ 3 là Menu Bill.

Trong Menu Bill có :

Cấp thứ 4 MenuXoa và MenuSua

**Lưu ý:** mỗi khi thực hiện chức năng thành công, sẽ hiển thị lại Menu hiện tại của nó.

## Class Diagram

Diagram

Description automatically generated

* Đây là một chương trình khá đơn giản, chỉ cần quản lý hóa đơn thanh toán, qua thảo luận và đưa ra kết quả của toán nhóm, nhóm chỉ sử dụng 2 class, class HoaDon và class Node. Class hóa đơn để quản lý dữ liệu cho hóa đơn, Class Node quản lý dự liệu hóa đơn trên một cấu trúc cây nhị phân.
* Chúng tôi chỉ sử dụng 1 trong 4 tính chất đặc trưng của lập trình hướng đối tượng đó là tính đóng gói dữ liệu.
* Chương trình này chúng em thực hiện 2 cấu trúc dữ liệu lưu trữ là:
  + Danh sách liên kết, dùng thư viện #include<list> có sẵn để thực hiện việc lưu khách vip
  + Cây, tự khởi tạo để lưu hóa đơn thanh toán, cũng như sử lí một số chức năng trên cấu trúc cây ấy.

## Giao diện App

* **Một số giao diện app mà nhóm thực hiện được.**
* Giao diện Menu Main

Text

Description automatically generated with medium confidence

* Giao diện Menu Pay

Text

Description automatically generated

* Giao diện Menu Bill

Text

Description automatically generated

* Giao diện Xem Danh Sách Hóa Đơn

A picture containing graphical user interface

Description automatically generated

* Giao diện Menu Vip

Text

Description automatically generated

* Giao diện Xem danh sách khách vip (khách hàng có bill bằng hoặc trên 100 triệu)

Text

Description automatically generated

# CÀI ĐẶT THỬ NGHIỆM

* Lựa trên các lý thuyết và giao diện đã nêu ra ở trên nhóm chúng em tiến hành tự động hóa, chương trình quản lý thanh toán cho cửa hàng xe máy.

## Cài đặt

## Chia cấu trúc file

* Để thực hiện ý tưởng trên nhóm quyết định chia file theo mô hình 3 lớp:
* Từ hệ thống gọi xuống layer thứ 1 (main.cpp)
* Layer thứ 1 gọi xuống layer thứ 2 (menu.h kết nối menu.cpp)
* Layer thứ 2 gọi xuống layer thứ 3 (DocGhiVip.h kết nối DocGhiVip.cpp & HoaDon.h kết nối với HoaDon.cpp)
* Từ layer thứ 3 gọi xuống file.txt lưu trữ dữ liệu đẩy trở về Layer thứ 1 để hiển thị ra giao diện xử lý.

Diagram

Description automatically generated

## Tiến hành cài đặt code

* **Main.cpp**

\* Include thư viện tự lập trình menu ở layer 2.

#include"Menu.h"

\* Gọi hàm Menu() trong hàm main để khởi chạy chương trình.

int main()

{

    Menu();

    system("pause");

    return 0;

}

* **Menu.h**

**\*** Include thư viện tự lập trình menu ở layer 3 và thư viện của môi trường hỗ trợ sẵn.

#include<iostream>

using namespace std;

#include"HoaDon.h"

#include"DocGhiVip.h"

\*

void menuMain();

void menuVip();

void menuPay();

void menuBill();

void Menu();

* **Menu.cpp**

**\*** Include thư viện tự lập trình, khởi tạo h0ut có giá trị HANDLE để hỗ trợ làm màu cho giao diện.

#include"HoaDon.h"

#include"DocGhiVip.h"

HANDLE hOut;

**\*** Hàm Menu Chính.

void menuMain() {

    hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

    SetConsoleTextAttribute(hOut, 3 << 10 | 1 << 4 | 11);

    cout << ("\t\t     #   #   #   #   #  # #      #       \n");

    cout << ("\t\t     #   # #   # ##  #  #   #   # #      \n");

    cout << ("\t\t     ##### #   # # # #  #    # #####     \n");

    cout << ("\t\t     #   # #   # #  ##  #   #  #   #     \n");

    cout << ("\t\t     #   #   #   #   #  # #    #   #     \n");

    cout << "\n";

    SetConsoleTextAttribute(hOut,2);

    cout << "============ CHUONG TRINH QUAN LY HOA DON CUA HANG XE MAY ============" << "\n\n";

    SetConsoleTextAttribute(hOut, 4);

    cout << "\t\t============MENU MAIN===========\n";

    SetConsoleTextAttribute(hOut, 6);

    cout << "\t\t1.Thanh toan." << endl;

    cout << "\t\t2.Danh sach khach hang VIP." << endl;

    cout << "\t\t3.Thoat." << endl;

    SetConsoleTextAttribute(hOut, 4);

    cout << "\t\t=================================\n";

    SetConsoleTextAttribute(hOut,2);

    cout << "============================= END ====================================" << "\n\n";

}

\* Hàm menu vip.

void menuVip()

{

    hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

    cout << "\n\n";

    SetConsoleTextAttribute(hOut, 4);

    cout << "\t\t\t==========MENU VIP========\n\n";

    SetConsoleTextAttribute(hOut, 6);

    cout << "\t\t\t1. Xem danh sach khach VIP." << endl;

    cout << "\t\t\t2. Quay ve menu chinh" << endl << endl;

    SetConsoleTextAttribute(hOut, 4);

    cout << "\t\t\t===========================\n";

    cout << "\n\n";

}

**\*** Hàm menu thanh toán.

void menuPay()

{

    hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

    cout << "\n\n";

    SetConsoleTextAttribute(hOut, 4);

    cout << "\t\t\t============MENU PAY============\n\n";

    SetConsoleTextAttribute(hOut, 6);

    cout << "\t\t\t1. Tao hoa don." << endl;

    cout << "\t\t\t2. Quay ve Menu Chinh." << endl << endl;

    SetConsoleTextAttribute(hOut, 4);

    cout << "\t\t\t================================\n";

    cout << "\n\n";

}

\* Hàm menu hóa đơn.

void menuBill()

{

    cout << "\n\n";

    hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

    SetConsoleTextAttribute(hOut, 4);

    cout << "\t\t\t=================MENU BILL==================\n\n";

    SetConsoleTextAttribute(hOut, 6);

    cout << "\t\t\t1. Xem hoa don." << endl;

    cout << "\t\t\t2. Doc hoa don." << endl;

    cout << "\t\t\t3. Them hoa don." << endl;

    cout << "\t\t\t4. Tim kiem hoa don theo ma khach hang." << endl;

    cout << "\t\t\t5. Xoa hoa don theo theo ma khach hang." << endl;

    cout << "\t\t\t6. Luu du lieu." << endl;

    cout << "\t\t\t7. Quay ve Muc Thanh Toan." << endl;

    cout << "\t\t\t8. Quay ve Menu Chinh." << endl << endl;

    SetConsoleTextAttribute(hOut, 4);

    cout << "\t\t\t============================================\n";

    cout << "\n\n";

}

\*Hàm menu toàn bộ chương trình .

void Menu()

{

    Tree T;

    HOADON x;

    T = NULL;

    cout << endl;

    cout << endl;

    hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

    const char\* f = "Ds.txt";

    const char\* v = "vip.txt";

    int flag;

MenuMain: {

    menuMain();

    SetConsoleTextAttribute(hOut, 5);

    cout << "\* Nhap So De Thao Tac: ";

    cin >> flag;

    }

        if (flag == 1)

        {

            int flag1;

            system("cls");

MenuPay: {

    menuPay();

    SetConsoleTextAttribute(hOut, 5);

    cout << "\* Nhap So De Thao Tac: ";

    cin >> flag1;

}

        if (flag1 == 1)

        {

            system("cls");

            int flag2;

MenuBill: {

    menuBill();

    SetConsoleTextAttribute(hOut, 5);

    cout << "\* Nhap So De Thao Tac: ";

    cin >> flag2;

}

        if (flag2 == 1)

        {

            SetConsoleTextAttribute(hOut, 7);

            system("cls");

            cout << endl;

            cout << "==================================================================DANH SACH HOA DON==============================================================" << endl;

            cout << endl;

            cout << setw(10) << "MaKH" << setw(15) << "TenKH" << setw(15) << "DiaChi" << setw(15) << "SDT" << setw(15) << "MaXe" << setw(15) << "TenXe" << setw(15) << "DonGia" << setw(15) << "SoLuong" << setw(15) << "ThanhTien" << endl;

            cout << "=================================================================================================================================================" << endl;

            LNR(T);

            system("pause");

            system("cls");

            goto MenuBill;

        }

        else if (flag2 == 2)

        {

            SetConsoleTextAttribute(hOut, 2);

            system("cls");

            cin.ignore();

            docfile(f, T);

            cout << "----Du Lieu Da Duoc Doc Len-----" << endl;

            system("pause");

            system("cls");

            goto MenuBill;

        }

        else if (flag2 == 3)

        {

            SetConsoleTextAttribute(hOut, 7);

            system("cls");

            cin.ignore();

            createTree(T);

            system("cls");

            goto MenuBill;

        }

        else if (flag2 == 4)

        {

            SetConsoleTextAttribute(hOut, 7);

            system("cls");

            char maKH[50];

            cin.ignore();

            cout << "\* Nhap Ma Khach Hang can tim: ";

            gets\_s(maKH);

            Tree P = searchHD(T, maKH);

            if (P != NULL)

            {

                int flag4;

            MenuSua: {

                system("cls");

                SetConsoleTextAttribute(hOut, 6);

                cout << "----------Tim Thay Hoa Don: " << maKH << "--------" << endl;

                cout << "1. Hien thi thong tin Hoa Don " << endl;

                cout << "2. Sua thong tin Hoa Don " << endl;

                cout << "3. Quay lai muc Bill " << endl;

                SetConsoleTextAttribute(hOut, 5);

                cout << "\* Nhap So De Thao Tac: ";

                cin >> flag4;

                }

            if (flag4 == 1)

            {

                SetConsoleTextAttribute(hOut, 7);

                system("cls");

                cout << endl;

                cout << setw(10) << "MaKH" << setw(15) << "TenKH" << setw(15) << "DiaChi" << setw(15) << "SDT" << setw(15) << "MaXe" << setw(15) << "TenXe" << setw(15) << "DonGia" << setw(15) << "SoLuong" << setw(15) << "ThanhTien" << endl;

                cout << "=================================================================================================================================================" << endl;

                LNR(P);

                system("pause");

                goto MenuSua;

            }

            else if (flag4 == 2)

            {

                SetConsoleTextAttribute(hOut, 7);

                system("cls");

                cout << "\* Nhap Thong Tin Can Sua" << endl;

                cin.ignore();

                suaHD(P);

                goto MenuSua;

            }

            else if (flag4 == 3)

            {

                system("cls");

                goto MenuBill;

            }

            }

            else

            {

                cout << "Khong Ton Tai Hoa Don Nay" << endl;

            }

            goto MenuBill;

        }

        else if (flag2 == 5)

        {

            system("cls");

            char maKH[50];

            cin.ignore();

            cout << "\* Nhap Ma Khach Hang can tim: ";

            gets\_s(maKH);

            Tree P = searchHD(T, maKH);

            if (P != NULL)

            {

                int flag5;

                system("cls");

                cout << endl;

                cout << setw(10) << "MaKH" << setw(15) << "TenKH" << setw(15) << "DiaChi" << setw(15) << "SDT" << setw(15) << "MaXe" << setw(15) << "TenXe" << setw(15) << "DonGia" << setw(15) << "SoLuong" << setw(15) << "ThanhTien" << endl;

                cout << "=================================================================================================================================================" << endl;

                LNR(P);

                SetConsoleTextAttribute(hOut, 6);

                cout << "----------------------------------------------------" << endl;

                cout << "1. Xoa Hoa Don " << maKH << endl;

                cout << "2. Huy Bo! va Quay lai Menu Chinh" << endl;

                cout << "----------------------------------------------------" << endl;

                SetConsoleTextAttribute(hOut, 5);

                cout << "Nhap So De Thao Tac: ";

                cin >> flag5;

                if (flag5 == 1)

                {

                    SetConsoleTextAttribute(hOut, 7);

                    system("cls");

                    cin.ignore();

                    int del = 1;

                    while (del)

                    {

                        if (P != NULL)

                        {

                            cout << "---Xoa Thanh Cong---" << endl;

                            del = delHD(T, P->key.MaHoaDon);

                        }

                        else

                        {

                            cout << "---Khong Co Du Lieu---" << endl;

                            del = 0;

                        }

                        system("Pause");

                        goto MenuBill;

                    }

                }

                if (flag5 == 2)

                {

                    system("cls");

                    goto MenuBill;

                }

            }

            else

            {

                cout << "---- Khong Ton Tai Ma Sinh Vien Nay----" << endl;

            }

            goto MenuBill;

        }

        else if (flag2 == 6)

        {

            SetConsoleTextAttribute(hOut, 2);

            system("cls");

            ghifile(f, T);

            ghifileVip(v, T);

            cout << "----Da Luu Du Lieu-----" << endl;

            system("pause");

            system("cls");

            goto MenuBill;

        }

        else if (flag2 == 7)

        {

            system("cls");

            goto MenuPay;

        }

        else if (flag2 == 8)

        {

            system("cls");

            goto MenuMain;

        }

        }

        else if (flag1 == 2)

        {

            system("cls");

            goto MenuMain;

        }

        }

        if (flag == 2)

        {

            int flag3;

            system("cls");

        MenuVip: {

            menuVip();

            SetConsoleTextAttribute(hOut, 5);

            cout << "\* Nhap So De Thao Tac: ";

            cin >> flag3;

            }

        if (flag3 == 1)

        {

            system("cls");

            SetConsoleTextAttribute(hOut, 2);

            cout << "==============DANH SACH KHACH HANG THAN THIET==============" << endl << endl;

            SetConsoleTextAttribute(hOut, 11);

            inDanhSachKhachHang(ReadFile("vip.txt"));

            SetConsoleTextAttribute(hOut, 2);

            cout << "============================================================" << endl << endl;

            SetConsoleTextAttribute(hOut, 7);

            system("pause");

            system("cls");

            goto MenuVip;

        }

        else if (flag3 == 2)

        {

            system("cls");

            goto MenuMain;

        }

        if (flag == 3)

        {

            return;

        }

    }

}

* **DocGhiVip.h**

\* Include các thư viện có sẵn để hỗ trợ lập trình và thư viện hóa đơn tự tạo, ở đây lưu trữ danh khách vip bằng danh sách liên kết dùng thư viện list có sẵn.

#pragma once

#include <iostream>

#include<stdio.h>

#include<iomanip>

#include <fstream>

#include<string>

using namespace std;

#include<list>

#include"HoaDon.h"

**\***

void  ghifileVip(const char\* filename, Tree& T);

bool kiemTraTonTaiFile(string FileName);

void inDanhSachKhachHang(list<string> ListCus);

list<string> ReadFile(string FileName);

* **DocGhiVip.cpp**

**\*** Include thư viện tự tạo **.**

#include"DocGhiVip.h"

\* Hàm kiểm tra sự tồn tại của file.txt.

bool kiemTraTonTaiFile(string FileName)

{

    fstream p;

    p.open(FileName, ios::in);

    if (p.fail())

        return false;

    else

        return true;

    p.close();

}

\* Hàm hiển thị danh sách khách vip theo danh sách liên kết ra màn hình.

void inDanhSachKhachHang(list<string> ListCus)

{

    int dem = 0;

    for (list<string>::iterator i = ListCus.begin(); i != ListCus.end(); i++)

    {

        cout <<"                        "<< \* i << "                  "<< endl;

        if (\*i != "")

        {

            dem++;

        }

    }

    cout << "        CUA HANG XE MAY HONDA CO " << dem << " KHACH HANG THAN THIET!" << endl;

    cout<< "                        XIN CHUC MUNG !!!" <<endl<<endl;

}

\*Hàm Đọc File theo dòng theo danh sách liên kết.

list<string> ReadFile(string FileName)

{

    list<string> listStr;

    fstream fst;

    if (kiemTraTonTaiFile(FileName))

    {

        fst.open(FileName, ios::in);

        do

        {

            string s = "";

            getline(fst, s);

            listStr.push\_back(s);

        } while (!fst.eof());

        fst.close();

    }

    else

    {

        cout << "\t +. File khong ton tai.\n \t +. Hay chon file khac. \n";

        system("pause");

    }

    return listStr;

}

\* Hàm ghi dữ liệu từ cây xuống file txt (khách hóa đơn bằng hoặc trên 100 triệu).

void  ghifileVip(const char\* filename, Tree& T)

{

    if (T != NULL)

    {

        FILE\* f = fopen(filename, "ab");

        if (tinhTien(T->key) >= 100) {

            fprintf(f, "%s-%s\n", T->key.HoTen, T->key.SDT);

        }

        ghifileVip(filename, T->Left);

        ghifileVip(filename, T->Right);

        fclose(f);

    }

}

* **HoaDon.h**

\* Include các thư viện do ide hỗ trợ.

#pragma once

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <string>

#include <iostream>

using namespace std;

#include <windows.h>

#include<iomanip>

**\*** Class hóa đơn.

class HOADON {

public:

    char MaHoaDon[50];

    char HoTen[50];

    char DiaChi[50];

    char MaXe[50];

    char TenXe[50];

    char SDT[20];

    int SoLuong;

    int DonGia;

};

**\***Class node của cây.

class Node

{

public:

    HOADON key;

    Node\* Left, \* Right;

};

typedef Node\* Tree;

**\***

int compare(HOADON x, HOADON y);

HOADON nhapHoaDon();

void xuatHoaDon(HOADON x);

double tinhTien(HOADON x);

int insertNode(Tree& T, HOADON x);

void createTree(Tree& T);

void LNR(Tree T);

Tree searchHD(Tree T, char MaKH[]);

Tree suaHD(Tree P);

int delHD(Tree& T, char MaKH[]);

void  ghifile(const char\* filename, Tree& T);

void docfile(const char\* filename, Tree& T);

* **HoaDon.cpp**

\*Include thư viện tự tạo.

#include"HoaDon.h"

\* Hàm so sánh 2 mã hóa đơn.

int compare(HOADON x, HOADON y)

{

    return strcmp(x.MaHoaDon, y.MaHoaDon);

}

\* Hàm tính tiền hóa đơn (số lượng xe mua \* đơn giá).

double tinhTien(HOADON x)

{

    return x.SoLuong \* x.DonGia;

}

\* Hàm nhập 1 hóa đơn.

HOADON nhapHoaDon()

{

    HOADON x;

    cout << "\* Nhap So Hoa Don (Q De Quay Lai): ";

    gets\_s(x.MaHoaDon);

    if (strcmp(x.MaHoaDon, "q") == 0 || strcmp(x.MaHoaDon, "Q") == 0)

    {

        return x;

    }

    cout << "\* Nhap Ten Khach Hang: ";

    gets\_s(x.HoTen);

    cout << "\* Nhap Dia Chi: ";

    gets\_s(x.DiaChi);

    cout << "\* Nhap So Dien Thoai: ";

    gets\_s(x.SDT);

    cout << "\* Nhap Ma Xe: ";

    gets\_s(x.MaXe);

    cout << "\* Nhap Ten Xe: ";

    gets\_s(x.TenXe);

    cout << "\* Nhap So Luong: "; cin >> x.SoLuong;

    cout << "\* Nhap Don Gia: "; cin >> x.DonGia;

    while (getchar() != '\n');

    return x;

}

\* Hàm xuất 1 hóa đơn.

void xuatHoaDon(HOADON x)

{

    if (tinhTien(x) >= 100)

    {

        cout << setw(10) << x.MaHoaDon << setw(15) << x.HoTen << setw(15) << x.DiaChi << setw(15) << x.SDT << setw(15) << x.MaXe << setw(15) << x.TenXe << setw(15) << x.DonGia << setw(15) << x.SoLuong << setw(15) << tinhTien(x) << setw(15) << " VIP" << endl;

    }

    else

        cout << setw(10) << x.MaHoaDon << setw(15) << x.HoTen << setw(15) << x.DiaChi << setw(15) << x.SDT << setw(15) << x.MaXe << setw(15) << x.TenXe << setw(15) << x.DonGia << setw(15) << x.SoLuong << setw(15) << tinhTien(x) << endl;

}

\* Thêm hóa đơn vào cấu trúc cây.

int insertNode(Tree& T, HOADON x)

{

    if (T != NULL) {

        if (compare(T->key, x) == 0)

            return -1;

        if (compare(T->key, x) > 0)

            return insertNode(T->Left, x);

        else if (compare(T->key, x) < 0)

            return insertNode(T->Right, x);

    }

    T = (Node\*)malloc(sizeof(Node));

    if (T == NULL)

        return 0;

    T->key = x;

    T->Left = T->Right = NULL;

    return 1;

}

\* Hàm nhập danh sách hóa đơn.

void createTree(Tree& T)

{

    HOADON x;

    while (1) {

        cout << "\t-------Nhap Hoa Don--------" << endl << endl;

        x = nhapHoaDon();

        if (strcmp(x.MaHoaDon, "q") == 0 || strcmp(x.MaHoaDon, "Q") == 0)

            break;

        int check = insertNode(T, x);

        if (check == -1)

            cout << "---Ma Sinh Vien Da Co---\n" << endl;

        else if (check == 0)

            cout << "---Bo Nho Day---\n" << endl;

        else cout << "---Them Thanh Cong---\n\n";

    }

}

\*Hàm duyệt cây theo trung tự (LNR).

void LNR(Tree T)

{

    if (T != NULL)

    {

        LNR(T->Left);

        xuatHoaDon(T->key);

        LNR(T->Right);

    }

}

\* Hàm tìm hóa đơn theo mã hóa đơn.

Tree searchHD(Tree T, char MaHD[])

{

    Node\* P = T;

    if (T != NULL)

    {

        if (strcmp(T->key.MaHoaDon, MaHD) == 0)

        {

            return P;

        }

        else if (strcmp(T->key.MaHoaDon, MaHD) > 0)

        {

            return searchHD(T->Left, MaHD);

        }

        else

            return searchHD(T->Right, MaHD);

    }

    else

        return NULL;

}

\* Hàm sửa hóa đơn.

Tree suaHD(Tree P)

{

    cout << "Sua Ten Khach Hang(Q De Quay Lai): ";

    cin.getline(P->key.HoTen, 50);

    if (strcmp(P->key.HoTen, "q") == 0 || strcmp(P->key.HoTen, "Q") == 0)

    {

        return P;

    }

    cout << "Sua SDT: ";

    cin >> P->key.SDT;

    while (getchar() != '\n');

    return P;

}

\*Hàm xóa hóa đơn.

int delHD(Tree& T, char MaHD[]) {

    if (T == NULL)

        return 0;

    else if (strcmp(T->key.MaHoaDon, MaHD) > 0)

        return delHD(T->Left, MaHD);

    else if (strcmp(T->key.MaHoaDon, MaHD) < 0)

        return delHD(T->Right, MaHD);

    else

    {

        Node\* P = T;

        if (T->Left == NULL)

            T = T->Right;

        else if (T->Right == NULL)

            T = T->Left;

        else

        {

            Node\* S = T, \* Q = S->Left;

            while (Q->Right != NULL) {

                S = Q;

                Q = Q->Right;

            }

            P->key = Q->key;

            S->Right = Q->Left;

            delete Q;

        }

    }

    return 1;

}

\* Hàm đọc file.txt hóa đơn.

void docfile(const char\* filename, Tree& T) {

    HOADON x;

    FILE\* f = fopen(filename, "rb");

    if (f != NULL)

    {

        while (!feof(f))

        {

            fscanf(f, "%s\n", &x.MaHoaDon);

            fscanf(f, "%s\n", &x.HoTen);

            fscanf(f, "%s\n", &x.DiaChi);

            fscanf(f, "%s\n", &x.SDT);

            fscanf(f, "%s\n", &x.MaXe);

            fscanf(f, "%s\n", &x.TenXe);

            fscanf(f, "%d\n", &x.SoLuong);

            fscanf(f, "%d\n", &x.DonGia);

            double tt = tinhTien(x);

            fscanf(f, "%lf\n\n", &tt);

            insertNode(T, x);

        }

        fclose(f);

    }

    else

    {

        cout << "!!--Thong Bao: Danh Sach Hoa Don Dang Rong, Xin Vui Long Nhap Hoa Don--!!" << "\n\n";

        createTree(T);

    }

}

\* Hàm ghi danh sách hóa đơn xuống file.txt.

void  ghifile(const char\* filename, Tree& T)

{

    if (T != NULL)

    {

        FILE\* f = fopen(filename, "ab");

        fprintf(f, "%s\n", T->key.MaHoaDon);

        fprintf(f, "%s\n", T->key.HoTen);

        fprintf(f, "%s\n", T->key.DiaChi);

        fprintf(f, "%s\n", T->key.SDT);

        fprintf(f, "%s\n", T->key.MaXe);

        fprintf(f, "%s\n", T->key.TenXe);

        fprintf(f, "%d\n", T->key.SoLuong);

        fprintf(f, "%d\n", T->key.DonGia);

        fprintf(f, "%lf\n\n", tinhTien(T->key));

        ghifile(filename, T->Left);

        ghifile(filename, T->Right);

        fclose(f);

    }

}

\* Tổ chức file.txt lưu hóa đơn.

Graphical user interface, text, application, email

Description automatically generated

\*Tổ chức file.txt lưu khách vip.

Graphical user interface, text, application

Description automatically generated

**(\*) Chú thích.**

## Thử nghiệm

* **Qua thử nghiệm rút ra được một số vấn đề:**
* Chương trình chỉ hỗ trợ trên các hệ điều hành windows 8, 10, 11 trở lên ở tất cả các phiên bản trong hệ điều hành này.
* Bởi vì ứng dụng cấu trúc cây thao tác trên hóa đơn mà thực hiện các chức năng như, duyệt, thêm, xóa, sửa, tìm kiếm rất nhanh.
* Về đọc file, ghi file tùy theo số lượng dữ liệu mà mức độ đọc ghi có thể tăng dần thời gian.
* Giao diện tuy có nhiều màu sắc nhưng cũng không giảm đi tốc độ xử lý của chương trình tự động hóa này.

# TỰ ĐÁNH GIÁ

## Những kết quả đạt được

* Tiếp cận được với bài toán thực tế.
* Xây dựng được một chương trình tự động hóa, quá trình thanh toán cho cửa hàng xe máy HonDa.
* Dùng 2 cấu trúc lưu trữ trong một chương trình là cây và danh sách liên kết.
* Sử dụng được các thuật toán đã học, áp dụng vào chương trình.

## Những mặt hạn chế

* Chương trình chưa hỗ trợ nhiều, mục quản lý, chức năng .
* Phần giao diện chỉ tạm chấp nhận được, cần phải đẹp hơn nữa.

## Hướng phát triển tiếp theo

* Khắc phục các mặt hạn chế của ứng dụng.
* Bổ sung nhiều tính năng, chức năng hơn.
* Cải thiện giao diện tốt hơn.
* Chúng em xin hứa sau đồ án sẽ hoàn tất các mục trên để sản phẩm được hoàn thiện áp dụng được trong thực tế, giúp ích cho các cửa hàng bán xe máy nói chung và các lĩnh vực khác nói riêng.