TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**

**CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT NÂNG CAO**

**ĐỀ TÀI: SỬ DỤNG THUẬT TOÁN QUICKSORT TRONG HỆ THỐNG QUẢN LÝ BÁN ĐỒ GỖ**

|  |  |
| --- | --- |
|  |  |
| |  |  | | --- | --- | | **Sinh viên thực hiện** | **: NGUYỄN CẢNH VINH** | | **Giảng viên hướng dẫn** | **: NGUYỄN THỊ THANH TÂN** | | | **Ngành** | **: CÔNG NGHỆ THÔNG TIN** | | | **Chuyên ngành** | **: CÔNG NGHỆ PHẦN MỀM** | | | **Lớp** | **: D13CNPM2** | | | **Khóa** | **: 2018-2023** | | |  |
|  |  |

***Hà Nội, tháng 06 năm 2020***

**PHIẾU CHẤM ĐIỂM**

Sinh viên thực hiện:

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Điểm** | **Chữ Ký** |
| Nguyễn Cảnh Vinh |  |  |

Giảng viên chấm:

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm : |  |  |

MỤC LỤC

[LỜI MỞ ĐẦU 1](#_Toc44169120)

[CHƯƠNG 1. GIỚI THIỆU THUẬT TOÁN QUICKSORT 2](#_Toc44169121)

[1. Tổng quan về thuật toán 2](#_Toc44169122)

[1.1. Thuật toán là gì? 2](#_Toc44169123)

[1.2. Tính chất của thuật toán 2](#_Toc44169124)

[1.3. Ví dụ: 2](#_Toc44169125)

[2. Khái niệm về thuật giải 3](#_Toc44169126)

[2.1. Thuật giải là gì? 3](#_Toc44169127)

[3. Giải thuật Quicksort 3](#_Toc44169128)

[3.1. Định nghĩa 3](#_Toc44169129)

[3.2. Ưu điểm 3](#_Toc44169130)

[3.3. Nhược điểm 4](#_Toc44169131)

[3.4. Tầm quan trọng của việc phân đoạn trong Quick Sort 4](#_Toc44169132)

[3.5. Thuật toán phân đoạn 4](#_Toc44169133)

[3.6. Quy trình của thuật toán 4](#_Toc44169134)

[3.7. Đánh giá về độ phức tạp của thuật toán 7](#_Toc44169135)

[CHƯƠNG 2. PHÂN TÍCH BÀI TOÁN 8](#_Toc44169136)

[2.1. Tầm quan trọng phần mềm vào bài toán quản lý 8](#_Toc44169137)

[2.2. Phân tích bài toán 8](#_Toc44169138)

[2.2.1. Mô tả thuật toán quickstort 8](#_Toc44169139)

[2.2.2. Phân tích các chức năng 10](#_Toc44169140)

[2.2.3. Cơ sở dữ liệu bài toán 12](#_Toc44169141)

[CHƯƠNG 3. CÀI ĐẶT THUẬT TOÁN 15](#_Toc44169142)

[3.1. Chức năng đăng nhập 15](#_Toc44169143)

[3.2. Chức năng hiển thị 16](#_Toc44169144)

[3.3. Chức năng thêm đồ gỗ 18](#_Toc44169145)

[3.4. Chức năng sửa danh mục đồ gỗ 20](#_Toc44169146)

[3.5. Chức năng xóa danh mục loại gỗ 22](#_Toc44169147)

[3.6. Tìm kiếm mã đồ gỗ 23](#_Toc44169148)

[3.7. Sắp xếp Nhập hàng 24](#_Toc44169149)

[KẾT LUẬN 29](#_Toc44169150)

[TÀI LIỆU THAM KHẢO 30](#_Toc44169151)

**DANH MỤC HÌNH ẢNH**

[Hình Ảnh 1. Thuật toán Quicksort 6](#_Toc44169166)

[Hình Ảnh 2. Kết quả chạy Dev C++ 7](#_Toc44169167)

[Hình Ảnh 3.Class 9](#_Toc44169168)

[Hình Ảnh 4. Quick Sort 10](#_Toc44169169)

[Hình Ảnh 5.Hiển thị 10](#_Toc44169170)

[Hình Ảnh 6.Diagrams 12](#_Toc44169171)

[Hình Ảnh 7.Bảng ID thể hiện tài khoản mật khẩu để đăng nhập vào hệ thống 13](#_Toc44169172)

[Hình Ảnh 8. DMDoGo thể hiện thông tin của các sản phẩm đồ gỗ 13](#_Toc44169173)

[Hình Ảnh 9. KhachHang thể hiện thông tin khách hàng 13](#_Toc44169174)

[Hình Ảnh 10. NhaCungCap thể hiện thông tin nhà cung cấp sản phẩm cho cửa hàng 14](#_Toc44169175)

[Hình Ảnh 11. NHAPHANG thể hiện thông tin hàng nhập vào 14](#_Toc44169176)

[Hình Ảnh 12. DonDatHang thể hiện thông tin đơn hàng đã đặt 14](#_Toc44169177)

[Hình Ảnh 13. HoaDon thể hiện thông tin hóa đơn mua bán 14](#_Toc44169178)

[Hình Ảnh 14. From Đăng Nhập 15](#_Toc44169179)

[Hình Ảnh 15. Đăng nhập thành công 16](#_Toc44169180)

[Hình Ảnh 16. Đăng nhập thất bại 16](#_Toc44169181)

[Hình Ảnh 17. Quản lý nhập hàng 17](#_Toc44169182)

[Hình Ảnh 18. Hiển thị đơn đặt hàng 18](#_Toc44169183)

[Hình Ảnh 19. Thêm đồ gỗ 19](#_Toc44169184)

[Hình Ảnh 20. Sửa thông tin đồ gỗ 21](#_Toc44169185)

[Hình Ảnh 21. Xóa danh mục đồ gỗ 22](#_Toc44169186)

[Hình Ảnh 22. Tìm kiếm mã đồ gỗ 23](#_Toc44169187)

[Hình Ảnh 23. Sắp xếp nhập hàng 26](#_Toc44169188)

[Hình Ảnh 24. Sắp xếp đơn nhập hàng theo số lượng nhập nhập hàng 26](#_Toc44169189)

[Hình Ảnh 25. Sắp xếp đơn đặt hàng theo giá bán 28](#_Toc44169190)

LỜI MỞ ĐẦU

Hiện nay, trong tất cả các lĩnh vực kinh tế, văn hóa, nông nghiệp ….công nghệthông tin thực sự gắn liền và góp phần làm tăng hiệu quả của các lĩnh vực đó, cụ thể đã có nhiều phần mềm phục vụ trong văn hóa nghệ thuật giải trí như phần mềm thống kê tin nhắn bình chọn, phần mềm xử lý ảnh, âm thanh; trong nông nghiệp thì người ta chế tạo các phương tiện, máy móc phục vụ cho mùa vụ…; trong trí tuệ nhân tạo có robot, máy móc tự động hóa… Vậy để có được các sản phẩm đó phải trải qua các bước khác nhau trong đó quan trọng nhất là ý tưởng, hay là cách thức cho sản phẩm đó hoạt động. Tóm lại là thuật toán

Trong báo cáo của em nghiên cứu về thuật toán nhằm sắp xếp các dữ liệu đưa vào. Đề tài em nghiên cứu là :”Sử dụng thuật toán Quicksort trong hệ thống quản lý bán đồ gỗ”.

**Lý do chọn đề tài:**

Để thuận tiện và giảm thời gian thao tác mà đặc biệt là tìm kiếm dữ liệu một cách dễ dàng và nhanh chóng, thông thường trước khi thao tác dữ liệu trên mảng, trên tập tin thì chúng đã có thứ tự. Do vậy thao tác sắp xếp dữ liệu là một trong những thao các cần thiết và thường gặp trong quá trình lưu trữ, quản lý dữ liệu.

Có rất nhiều phương pháp sắp xếp nhưng ở đây chúng ta cùng tìm hiểu phương pháp sắp xếp nhanh Quicksort và nó được áp dụng sao vào quản lý hệ thống bán đồ gỗ.

CHƯƠNG 1. GIỚI THIỆU THUẬT TOÁN QUICKSORT

1. Tổng quan về thuật toán
   1. Thuật toán là gì?

Thuật toán là một tập hợp hữu hạn hay một dãy các qui tắc chặt chẽ của các chỉ thị, phương cách hay 1 trình tự các thao tác trên một đối tượng cụ thể được xác định và định nghĩa rõ ràng cho việc hoàn tất một số sự việc từ một trạng thái ban đầu cho trước; khi các chỉ thị này được áp dụng triệt để thì sẽ dẫn đến kết quả sau cùng như đã dự đoán trước.

Nói cách khác, thuật toán là một bộ các quy tắc hay quy trình cự thể nhằm giải quyết một vấn đề nào đó trong một số bước hữu hạn, hoặc nhằm cung cấp một kết quả từ một tâp hợp các dữ kiện đưa vào.

* 1. Tính chất của thuật toán
* Tính chính xác: để đảm bảo kết quả tính toán hay các thao tác mà máy tính thực hiện được là chính xác.
* Tính rõ ràng: Thuật toán phải được thể hiện bằng các câu lệnh minh bạch, các câu lệnh đợc sắp xếp theo thứ tự nhất định
* Tính khách quan: Một thuật toán dù được viết bởi nhiều người trên nhiều máy tính vẫn phải cho kết quả như nhau.
* Tính phổ dụng: Thuật toán không chỉ áp dụng cho một bài toán nhất định mà có thể áp dụng cho một lớp các bài toán có đầu vào tương tự nhau.
* Tính kết thúc: Thuật toán phải gồm một số hữu hạn các bước tính toán.
  1. Ví dụ:

Thuật toán để giải được phương trình bậc nhất P(x): *a*x + *b* = *c*, (*a*, *b*, *c* là các số thực), trong tập hợp các số thực có thể là một bộ các bước sau đây:

1. Nếu a = 0
   * b = c thì P(x) có nghiệm bất kỳ
   * b ≠ c thì P(c) vô nghiệm
2. Nếu a ≠ 0
   * P(x) có duy nhất một nghiệm x = (c-b)/a
3. Khái niệm về thuật giải
   1. Thuật giải là gì?

Giải thuật là một tập hợp hữu hạn các chỉ thị để được thực thi theo một thứ tự nào đó để thu được kết quả mong muốn. Nói chung thì giải thuật là độc lập với các ngôn ngữ lập trình, tức là một giải thuật có thể được triển khai trong nhiều ngôn ngữ lập trình khác nhau.

1. Giải thuật Quicksort
   1. Định nghĩa

Quick sort là thuật toán sắp xếp được phát minh lần đầu bởi C.A.Hoare vào năm 1960.

Quick Sort là thuật toán sắp xếp hiệu quả dựa trên việc phân chia mảng dữ liệu thành các phần tử nhỏ hơn.

Giải thuật sắp xếp nhanh chia mảng thành hai phần bằng cách so sanh từng phần từ của mảng với một phần tử được gọi là phần tử chốt.

* 1. Ưu điểm
* Dễ cài đặt.
* Áp dụng rộng dãi và hiệu quả trong hầu hết các trường hợp
* Tiêu tốn ít tài nguyên hơn do so với các giải thuật khác.
  1. Nhược điểm
* Phải cài đặt bằng đệ quy (có thể không dùng đệ quy nhưng việc cài đặt sẽ phức tạp hơn nhiều)
* Yêu cầu độ chính xác cực kỳ cao.
  1. Tầm quan trọng của việc phân đoạn trong Quick Sort

Mấu chốt chính của thuật toán quick sort là việc phân đoạn dãy số

Mục tiêu của công việc này là:

Cho một mảng và một phần tử x là pivot.

Đặt x vào đúng vị trí của mảng đã sắp xếp.

Di chuyển tất cả các phần tử của mảng mà nhỏ hơn x sang bên trái vị trí của x, và di chuyển tất cả các phần tử của mảng mà lớn hơn x sang bên phải vị trí của x.

Khi đó ta sẽ có 2 mảng con: mảng bên trai của x và mảng bên phải của x. Tiếp tục công việc với mỗi mảng con (chọn pivot, phân đoạn) cho tới khi mảng được sắp xếp.

* 1. Thuật toán phân đoạn

Đặt pivot là phần tử cuối cùng của dãy số arr. Chúng ta bắt đầu từ phần tử trái nhất của dãy số có chỉ số là left, và phần tử phải nhất của dãy số có chỉ số là right -1(bỏ qua phần tử pivot). Chừng nào left < right mà arr[left] > pivot và arr[right] < pivot thì đổi chỗ hai phần tử left và right. Sau cùng, ta đổi chỗ hai phần tử left và pivot cho nhau. Xem hình minh họa phía dưới. Khi đó, phần tử left đã đứng đúng vị trí và chia dãy số làm đôi (bên trái và bên phải)

* 1. Quy trình của thuật toán

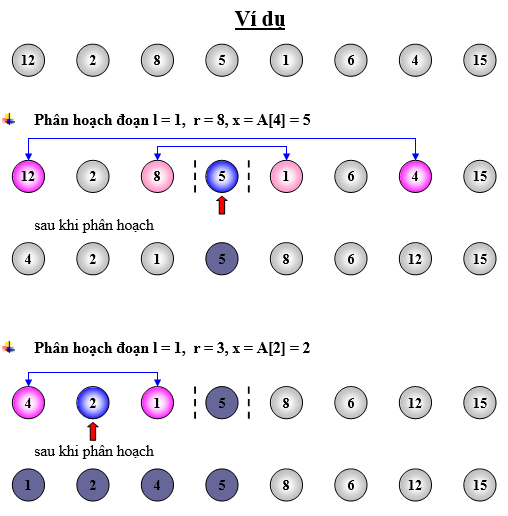
Bước 1: Lấy phần tử chốt là phần tử ở cuối danh sách.

Bước 2: Chia mảng theo phần tử chốt.

Bước 3: Sử dụng sắp xếp nhanh một cách đệ qui với mảng con bên trái.

Bước 4: Sử dụng sắp xếp nhanh một cách đệ qui với mảng con bên phải. duk

Ví dụ:

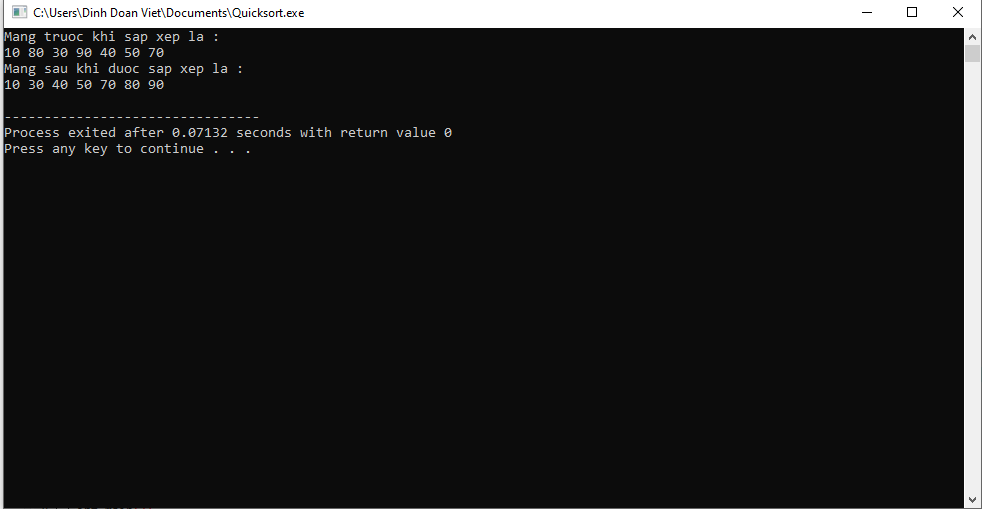


Hình Ảnh 1. Thuật toán Quicksort

**-** Ở đây chúng ta sẽ sắp xếp 1 mảng số nguyên:

|  |
| --- |
| #include<iostream>  using namespace std;  void QuickSort (int arr[], int left, int right)  { if (left >= right) return;  int pivot = arr[(left+right)/2];  int i = left, j = right;  while (i<j){  while (arr[i]>pivot) i++;  while (arr[j]<pivot) j--;  if (i <= j){  int tam = arr[i];  arr[i] = arr[j];  arr[j] = tam;  i++;  j--;}  }  if (left < j) QuickSort(arr, left, j);  if (i < right) QuickSort(arr, i, right);}  void Xuat(int arr[], int size){  for (int i=0; i < size; i++)  printf("%d ", arr[i]);  printf("\n");}  int main(){  int arr[] = {10, 80, 30, 90, 40, 50, 70};  cout << "Mang truoc khi sap xep la : \n";  int n = sizeof(arr)/sizeof(arr[0]);  Xuat(arr, n);  QuickSort(arr, 0, n-1);  cout <<"Mang sau khi duoc sap xep la : \n";  Xuat(arr, n);  return 0; } |

* **Kết quả chạy trên Dev C++ như sau :**



Hình Ảnh 2. Kết quả chạy Dev C++

* 1. Đánh giá về độ phức tạp của thuật toán

Độ phức tạp của thuật toán Quick Sort:

* + - Trường hợp tốt: O(nlog(n))
    - Trung bình: O(nlog(n))
    - Trường hợp xấu: O(n^2)

Không gian bộ nhớ sử dụng: O(log(n))

CHƯƠNG 2. PHÂN TÍCH BÀI TOÁN

2.1. Tầm quan trọng phần mềm vào bài toán quản lý

Ngày nay quản lý tốt nguồn nhân sự là một vấn đề mang tính sống còn với các doanh nghiệp vì nó ảnh hưởng trực tiếp đến hiệu quả kinh doanh và sự phát triển bền vững của tổ chức . Đó không những là đòi hỏi cho các doanh nghiệp lớn , các tổng công ty hay các công ty đa quốc gia mà còn cho cả các cơ quan , các doanh nghiệp vừa và nhỏ đang ngày càng bài bản và chuyên nghiệp hơn . Ứng dụng CNTT trong quản lý nhân sự tại Việt Nam không phải là một bài toán mới . Doanh nghiệp , Cơ quan có thể lựa chọn một hệ thống chuyên biệt hoặc tích hợp , nổi tiếng hay vô danh , đóng gói sẵn hay tự xây dựng , tuỳ theo ngân sách và yêu cầu đặc thù của họ . Chính vì vậy sẽ không có một giải pháp nào là tối ưu cho mọi doanh nghiệp , mọi cơ quan . | Phần mềm của em là phần mềm quản lý nhân sự trong trung tâm tính toán - Tổng cục thống kê , phần mềm được xây dựng dựa trên nền tảng công nghệ Winform 2019, và vận hành trên hệ quản trị cơ sở dữ liệu nền là Microsoft SQL Server 2019 . Phần mềm sẽ giải quyết được những khó khăn trong quản lý nhân sự và tính lương trong trung tâm . Cụ thể , phần mềm sẽ lưu trữ xử lý lý lịch chi tiết về sản phẩm, hóa đơn, khách hàng, nhập hàng, thu chi.

2.2. Phân tích bài toán

2.2.1. Mô tả thuật toán quickstort

Sau mỗi giao dịch khách hàng sẽ điền vào phiếu đặt hàng số lượng sản phẩm muốn mua và nhận về hóa đơn thanh toán. Dựa vào các số liệu đó chúng ta sẽ đánh giá cũng như sắp xếp giá cả để xác định thu chi của cửa hàng làm sao không bị lỗ.

Input: Danh sách đơn đặt hàng

Output: Danh sách đơn đặt hàng giá từ cao xuống thấp

Các bước thực hiện bài toán:

Đầu tiên chúng ta cần khai báo một class DonDatHang để lưu danh sách đầu vào

|  |
| --- |
| class DonDatHangSX  {  private string MADDH;  private DateTime NGAYDH;  private string MADG;  private float GIA;  private string MAKH;  public DonDatHangSX(DataRow row)  {  this.MADDH = row["MaDDH"].ToString();  this.NGAYDH = (DateTime)row["NgayDH"];  this.MADG = row["MaDG"].ToString();  this.GIA = (float)Convert.ToDouble(row["Gia"].ToString());  this.MAKH = row["MaKH"].ToString();  }  public string MaDDH1 { get => MADDH; set => MADDH = value; }  public DateTime NgayDH1 { get => NGAYDH; set => NGAYDH = value; }  public string MaDG1 { get => MADG; set => MADG = value; }  public float GIA1 { get => GIA; set => GIA = value; }  public string MaKH1 { get => MAKH; set => MAKH = value; }  } |

Hình Ảnh 3.Class

Sau đó ta cần sử dụng quicksort để sắp xếp dữ liệu đã đc lưu vào Class

|  |
| --- |
| List<DonDatHangSX> QuickSort(List<DonDatHangSX> a, int left, int right)  {  int i, j;  float x;  x = a[(left + right) / 2].GIA1;  i = left; j = right;  while (i < j)  {  while (a[i].GIA1 > x) i++;  while (a[j].GIA1 < x) j--;  if (i <= j)  {  DonDatHangSX tam;  tam = a[i];  a[i] = a[j];  a[j] = tam;  i++;  j--;  }  }  if (left < j)  QuickSort(a, left, j);  if (i < right)  QuickSort(a, i, right);  return a;  } |

Hình Ảnh 4. Quick Sort

Cuối cùng ta lưu lại dữ liệu từ class vào lại datatable và hiển thị kết quả

|  |
| --- |
| void HienThi1()  {  string sqlconn = @"Data Source=DESKTOP-Q6N4J46\SQLEXPRESS01;Initial Catalog=QLCuaHangDoGo;Integrated Security=True";  SqlConnection conn = new SqlConnection(sqlconn);  conn.Open();  string cmd = "SELECT \* FROM DonDatHang ";  SqlCommand scmd = new SqlCommand(cmd);  List<DonDatHangSX> listnv = new List<DonDatHangSX>();  DataTable data = new DataTable();  SqlDataAdapter da = new SqlDataAdapter(cmd, conn);  da.Fill(data);  foreach (DataRow items in data.Rows)  {  DonDatHangSX nhaphang = new DonDatHangSX(items);  listnv.Add(nhaphang);  }  conn.Close();  dgvDonDatHang.DataSource = QuickSort(listnv, 0, listnv.Count - 1);  } |

Hình Ảnh 5.Hiển thị

2.2.2. Phân tích các chức năng

**Chức năng đăng nhập**

Chức năng này diễn tả một người dùng đăng nhập vào hệ thống như thế nào.

Chức năng này bắt đầu người dùng muốn đăng nhập vào hệ thống quản lý.

1. Hệ thống yêu cầu người dùng nhập tài khoản và mật khẩu của mình.
2. Người dùng nhập tài khoản và mật khẩu.
3. Hệ thống xác thực tài khoản và mật khẩu nhập vào và cho phép người dùng vào hệ thống.

**Chức năng quản lý**

1-Use case cho phép nhà quản lý thực hiện chức năng quản lý thông tin nhân viên làm việc trong rạp.

2-Use case có 7 chức năng chính: Hiển thị thông tin đơn đặt hàng, Thêm đơn đặt hàng, Cập nhật thông tin đặt hàng, Xóa thông tin đặt hàng, Tìm kiếm đơn đặt hàng, Sắp Xếp hóa đơn theo giá . Mỗi chức năng này nhà quản lý sử dụng khi muốn thực hiện những hành động tương tự trên thực tế trong công tác quản lý hóa đơn.

**Hiển thị đơn nhập hàng, khách hàng, danh mục đồ gỗ,…**

Phầm mềm hiển thị tất cả các đơn đặt hàng, danh mục đồ gỗ,… từ cơ sở dữ liêu có trong cửa hàng dưới dạng dataview

**Thêm đơn đặt hàng**

* Hệ thống yêu cầu nhà quản lý nhập vào thông tin của đơn hàng muốn thêm vào. Thông tin bao gồm

+ Mã đơn đặt hàng:

+ Ngày đặt hàng:

+ Mã đồ gỗ

+ Giá đặt hàng:

+ Mã khách hàng:

Mỗi lần bạn quản lý cung cấp thông tin yêu cầu, hệ thống tổng hợp. đơn hàng mới sẽ được thêm vào hệ thống.

**Sửa thông tin đơn hàng**

1.Hệ thống yêu cầu nhà quản lý nhập vào thông tin của đơn hàng muốn sửa vào. Thông tin bao gồm

+ Mã đơn đặt hàng:

+ Ngày đặt hàng:

+ Mã đồ gỗ

+ Giá đặt hàng:

+ Mã khách hàng:

2.Mỗi lần Ban quản lý nhập thông tin cần thiết, hệ thống sẽ cập nhật bản ghi đơn hàng đó.

**Xóa đơn hàng**

1. Hệ thống yêu cầu Ban quản lý nhập mã số đơn hàng.

2. Ban quản lý nhập mã đơn đặt hàng, hệ thống tìm và hiển thị thông tin đơn đặt hàng.

3. Hệ thống nhắc Ban quản lý xác nhận việc xóa đơn hàng đó.

4. Ban quản lý xác nhận xóa.

5. Hệ thống xóa đơn đặt hàng khỏi hệ thống.

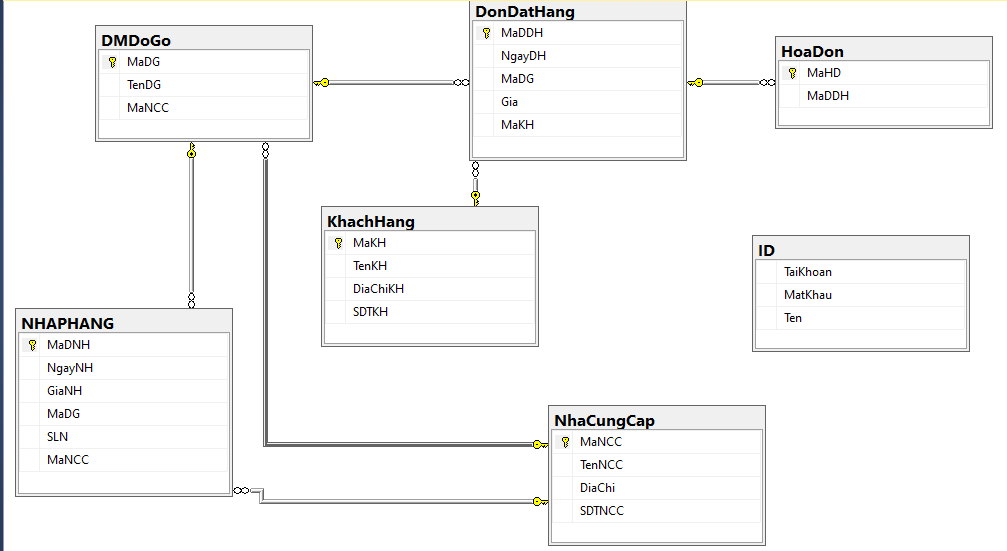
**Tìm Kiếm**

Bạn nhập dữ liệu mã đơn đặt hàng cần tìm kiếm vào ô tim kiếm phần mềm sẽ tìm kiếm những dữ liệu đơn đặt hàng có mã như vậy cho bạn

**Sắp xếp đơn đặt hàng theo giá đồ gỗ**

Dựa vào thuật toán quickstort hệ thống sắp xếp đơn đặt hàng theo hướng là giá cao tới thấp.

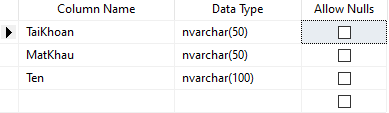
2.2.3. Cơ sở dữ liệu bài toán



Hình Ảnh 6.Diagrams

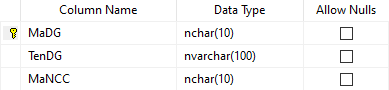
Bài toán của chúng ta gồm 4 bảng:

Bảng 2.1:Bảng ID



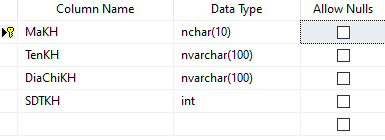
Hình Ảnh 7.Bảng ID thể hiện tài khoản mật khẩu để đăng nhập vào hệ thống

Bảng 2.2: Bảng DMDoGo



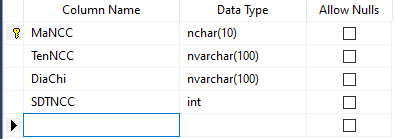
Hình Ảnh 8. DMDoGo thể hiện thông tin của các sản phẩm đồ gỗ

Bảng 2.3:Bảng KhachHang



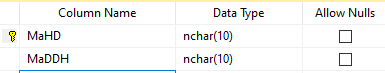
Hình Ảnh 9. KhachHang thể hiện thông tin khách hàng

Bảng 2.4:Bảng NhaCungCap



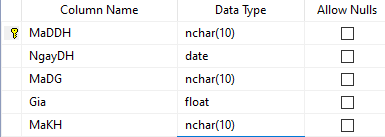
Hình Ảnh 10. NhaCungCap thể hiện thông tin nhà cung cấp sản phẩm cho cửa hàng

Bảng 2.5: Bảng NHAPHANG

****

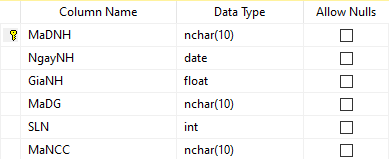
Hình Ảnh 11. NHAPHANG thể hiện thông tin hàng nhập vào

Bảng 2.6: Bảng DonDatHang

******

Hình Ảnh 12. DonDatHang thể hiện thông tin đơn hàng đã đặt

Bảng 2.7: Bảng DonHang

****

Hình Ảnh 13. HoaDon thể hiện thông tin hóa đơn mua bán

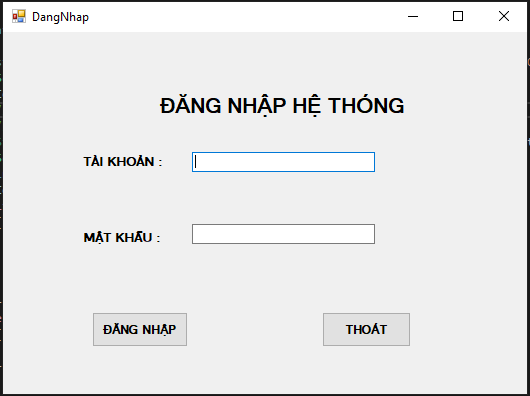
CHƯƠNG 3. CÀI ĐẶT THUẬT TOÁN

3.1. Chức năng đăng nhập

Chức năng đăng nhập để đăng nhập vào hệ thống quản lý bán đồ gỗ

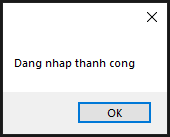
CODE:

|  |
| --- |
| private void btnDN\_Click(object sender, EventArgs e){  string Conn\_tr = @"Data Source=DESKTOP-Q6N4J46\SQLEXPRESS01;Initial Catalog=QuanLyNhanVien;Integrated Security=True";  SqlConnection Conn = new SqlConnection(Conn\_tr);  Conn.Open();  String Query = "select COUNT(\*) from NguoiDung where TaiKhoan='" + txtTK.Text + "' and MatKhau='" + txtMK.Text + "'";  SqlCommand Cmd = new SqlCommand(Query, Conn);  int SoLuong = int.Parse(Cmd.ExecuteScalar().ToString());  Conn.Close();  if (SoLuong == 1){  MessageBox.Show("Dang nhap thanh cong");  Form1 f = new Form1();  f.Show();  this.Hide();  }  else {  MessageBox.Show("Dang nhap khong thanh cong");  }  } |



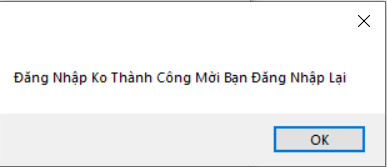
Hình Ảnh 14. From Đăng Nhập

Đăng nhập thành công



Hình Ảnh 15. Đăng nhập thành công

Đăng nhập thất bại



Hình Ảnh 16. Đăng nhập thất bại

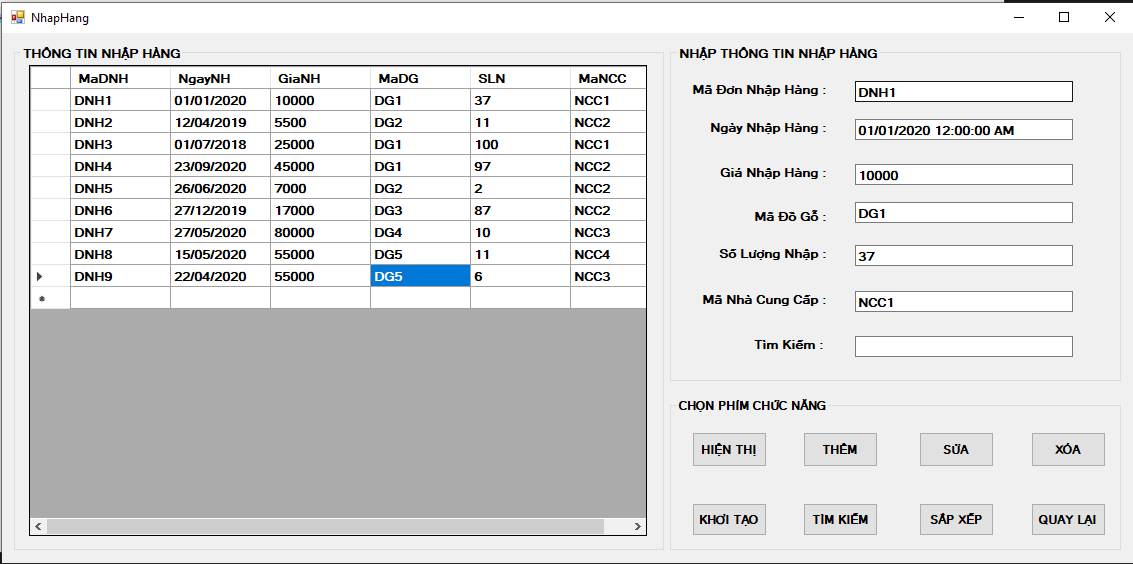
3.2. Chức năng hiển thị

Hiển thị toàn bộ Đơn nhập hàng, bảng nhập hàng…

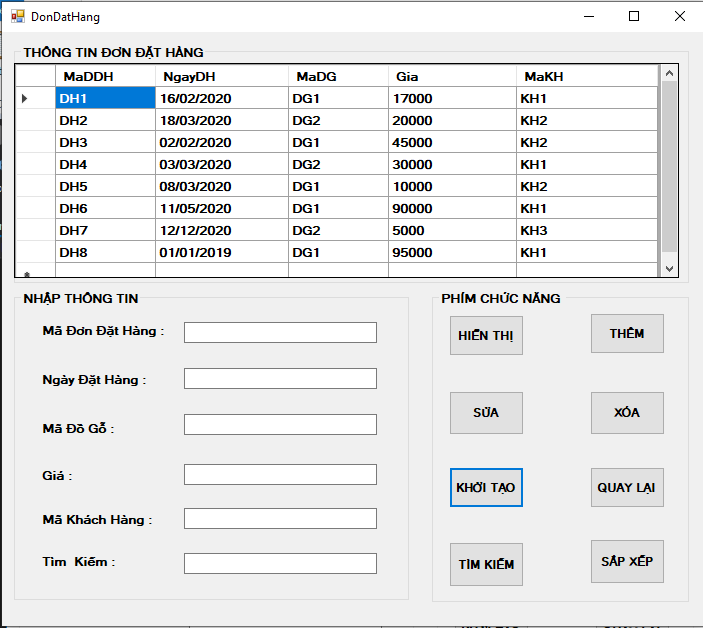
CODE:

|  |
| --- |
| public void HienThi()  {  string Conn\_tr = @"Data Source=DESKTOP-Q6N4J46\SQLEXPRESS01;Initial Catalog=QLCuaHangDoGo;Integrated Security=True ";  SqlConnection Conn = new SqlConnection(Conn\_tr);  Conn.Open();  string Query = "Select \*From NhapHang";  SqlCommand Cmd = new SqlCommand(Query, Conn);  SqlDataAdapter da = new SqlDataAdapter(Cmd);  DataSet ds = new DataSet();  da.Fill(ds);  dgvNhapHang.DataSource = ds.Tables[0];  Conn.Close();  } |

|  |
| --- |
| public void HienThi()  {  string Conn\_tr = @"Data Source=DESKTOP-Q6N4J46\SQLEXPRESS01;Initial Catalog=QLCuaHangDoGo;Integrated Security=True";  SqlConnection Conn = new SqlConnection(Conn\_tr);  Conn.Open();  string Query = "select \*from DonDatHang";  SqlCommand Cmd = new SqlCommand(Query, Conn);  SqlDataAdapter da = new SqlDataAdapter(Cmd);  DataSet ds = new DataSet();  da.Fill(ds);  dgvDonDatHang.DataSource = ds.Tables[0];  Conn.Close();  } |



Hình Ảnh 17. Quản lý nhập hàng



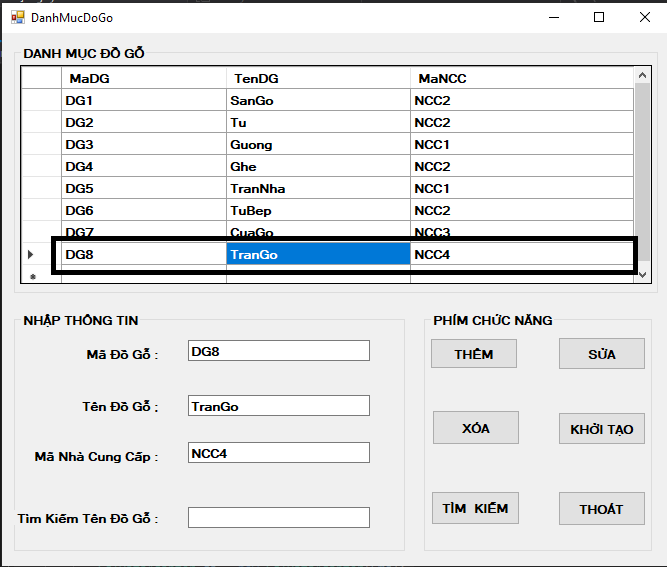
Hình Ảnh 18. Hiển thị đơn đặt hàng

3.3. Chức năng thêm đồ gỗ

Thêm mới một loại sản phẩm vào hệ thống

CODE:

|  |
| --- |
| private void btnThem\_Click(object sender, EventArgs e)  {  try  {  string Conn\_tr = @"Data Source=DESKTOP-Q6N4J46\SQLEXPRESS01;Initial Catalog=QLCuaHangDoGo;Integrated Security=True";  SqlConnection Conn = new SqlConnection(Conn\_tr);  Conn.Open();  string Query = "Insert into DanhMucDoGo Values('" + txtMaDG.Text + "','" + txtTenDG.Text + "','" + txtMaNCC.Text + "')";  SqlCommand Cmd = new SqlCommand(Query, Conn);  Cmd.ExecuteNonQuery();  Conn.Close();  MessageBox.Show("Them thanh cong");  HienThi();  }  catch  {  MessageBox.Show("Them khong thanh cong, moi ban nhap lai!!!");  }  } |



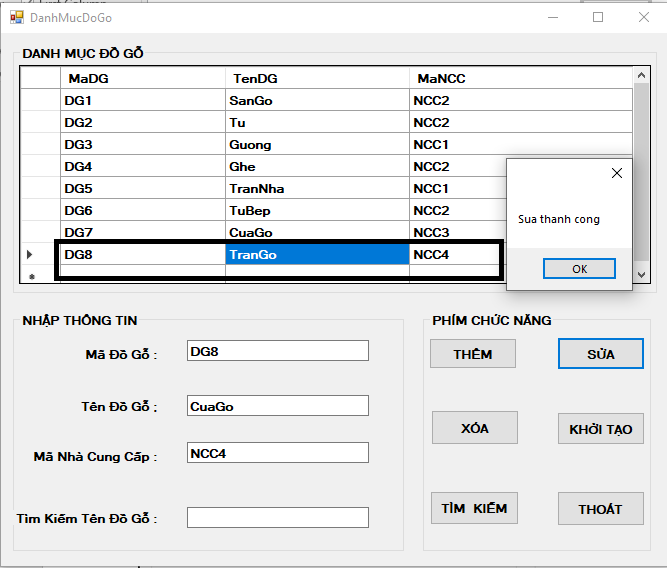
Hình Ảnh 19. Thêm đồ gỗ

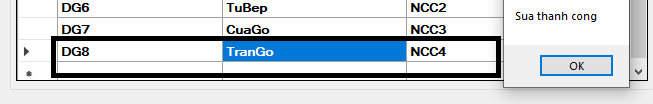
3.4. Chức năng sửa danh mục đồ gỗ

Chức năng này cho phép sửa thông tin đồ gỗ trong cửa hàng

CODE:

|  |
| --- |
| private void btnSua\_Click(object sender, EventArgs e)  {  string Conn\_tr = @"Data Source=DESKTOP-Q6N4J46\SQLEXPRESS01;Initial Catalog=QLCuaHangDoGo;Integrated Security=True";  SqlConnection Conn = new SqlConnection(Conn\_tr);  Conn.Open();  string Query = "Update DanhMucDoGo set TenDG='"+txtTenDG.Text+"',MaNCC='"+txtMaNCC.Text+"' where MaDG='"+txtMaDG.Text+"'";  SqlCommand Cmd = new SqlCommand(Query, Conn);  Cmd.ExecuteNonQuery();  Conn.Close();  MessageBox.Show("Sua thanh cong");  HienThi();  } |





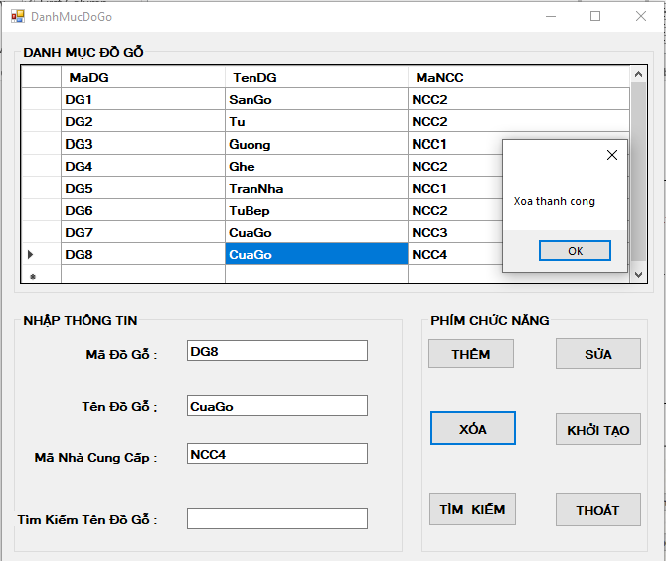
Hình Ảnh 20. Sửa thông tin đồ gỗ

3.5. Chức năng xóa danh mục loại gỗ

Chức năng này giúp xóa dữ liệu một loại đồ gỗ trong cửa hàng

CODE:

|  |
| --- |
| private void tnXoa\_Click(object sender, EventArgs e){  string Conn\_tr = @"Data Source=DESKTOP-Q6N4J46\SQLEXPRESS01;Initial Catalog=QLCuaHangDoGo;Integrated Security=True";  SqlConnection Conn = new SqlConnection(Conn\_tr);  Conn.Open();  string Query = "Delete From DanhMucDoGo where MaDG='"+txtMaDG.Text+"'";  SqlCommand Cmd = new SqlCommand(Query, Conn);  Cmd.ExecuteNonQuery();  Conn.Close();  MessageBox.Show("Xoa thanh cong");  HienThi();  } |



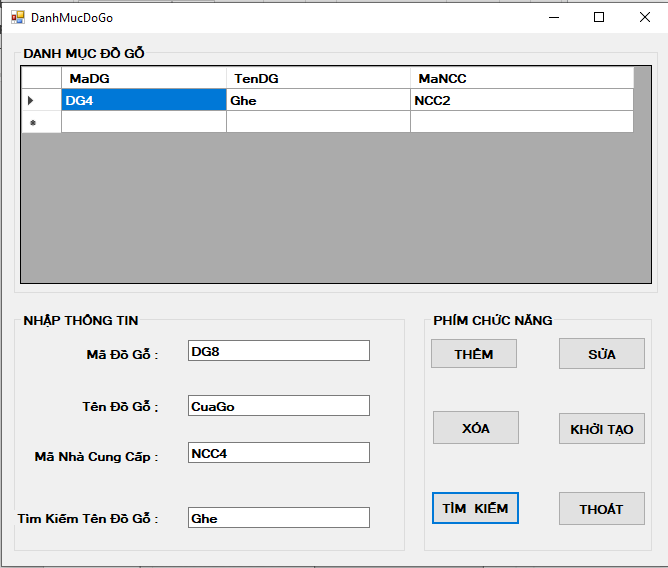
Hình Ảnh 21. Xóa danh mục đồ gỗ

3.6. Tìm kiếm mã đồ gỗ

Chức năng này giúp ta tìm kiếm đồ gỗ theo tên hoặc mã đồ gỗ giúp việc tìm đồ gỗ trở lên dễ dàng hơn

CODE:

|  |
| --- |
| private void btnTimKiem\_Click(object sender, EventArgs e){  string Conn\_tr = @"Data Source=DESKTOP-Q6N4J46\SQLEXPRESS01;Initial Catalog=QLCuaHangDoGo;Integrated Security=True";  SqlConnection Conn = new SqlConnection(Conn\_tr);  Conn.Open();  string TenDG = txtTK.Text;  if (TenDG.Length == 0){  HienThi();  }  else{  string Query = "select \*From DanhMucDoGo where TenDG='"+TenDG+"'";  SqlCommand Cmd = new SqlCommand(Query, Conn);  SqlDataAdapter da = new SqlDataAdapter(Cmd);  DataSet ds = new DataSet();  da.Fill(ds);  dgvDMDG.DataSource = ds.Tables[0];  }  Conn.Close();  } |



Hình Ảnh 22. Tìm kiếm mã đồ gỗ

3.7. Sắp xếp Nhập hàng

Chức năng này giúp chúng ta sắp xếp bảng nhập hàng theo giá nhập hàng , Số lượng đơn hàng

|  |
| --- |
| List<LopNhapHangSX> QuickSort(List<LopNhapHangSX> a, int left, int right)  {  int i, j;  float x;  x = a[(left + right) / 2].SLN1;  i = left; j = right;  while (i < j)  {  while (a[i].SLN1 > x) i++;  while (a[j].SLN1 < x) j--;  if (i <= j)  {  LopNhapHangSX tam;  tam = a[i];  a[i] = a[j];  a[j] = tam;  i++;  j--;  }  }  if (left < j)  QuickSort(a, left, j);  if (i < right)  QuickSort(a, i, right);  return a;  } |

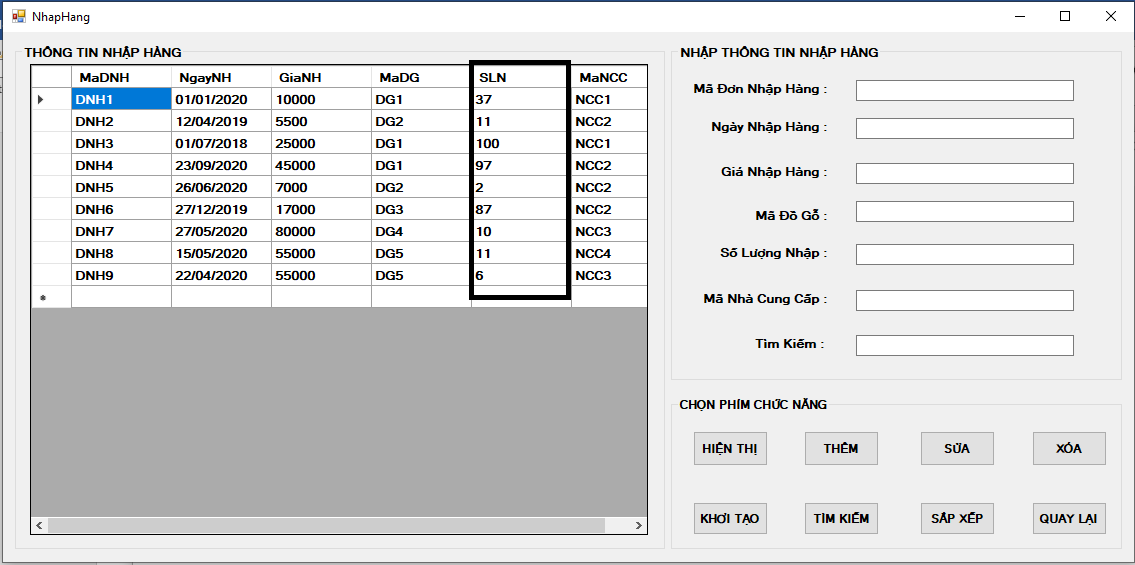
|  |
| --- |
| List<DonDatHangSX> QuickSort(List<DonDatHangSX> a, int left, int right)  {  int i, j;  float x;  x = a[(left + right) / 2].GIA1;  i = left; j = right;  while (i < j)  {  while (a[i].GIA1 > x) i++;  while (a[j].GIA1 < x) j--;  if (i <= j)  {  DonDatHangSX tam;  tam = a[i];  a[i] = a[j];  a[j] = tam;  i++;  j--;  }  }  if (left < j)  QuickSort(a, left, j);  if (i < right)  QuickSort(a, i, right);  return a;  } |

**Lưu lại dữ liệu từ class vào lại datatable và hiển thị kết quả**

|  |
| --- |
| void HienThi1()  {  string sqlconn = @"Data Source=DESKTOP-Q6N4J46\SQLEXPRESS01;Initial Catalog=QLCuaHangDoGo;Integrated Security=True";  SqlConnection conn = new SqlConnection(sqlconn);  conn.Open();  string cmd = "Select \*From NhapHang";  SqlCommand scmd = new SqlCommand(cmd);  List<LopNhapHangSX> listnv = new List<LopNhapHangSX>();  DataTable data = new DataTable();  SqlDataAdapter da = new SqlDataAdapter(cmd, conn);  da.Fill(data);  foreach (DataRow items in data.Rows)  {  LopNhapHangSX nhaphang = new LopNhapHangSX(items);  listnv.Add(nhaphang);  }  conn.Close();  dgvNhapHang.DataSource = QuickSort(listnv, 0, listnv.Count - 1);  } |

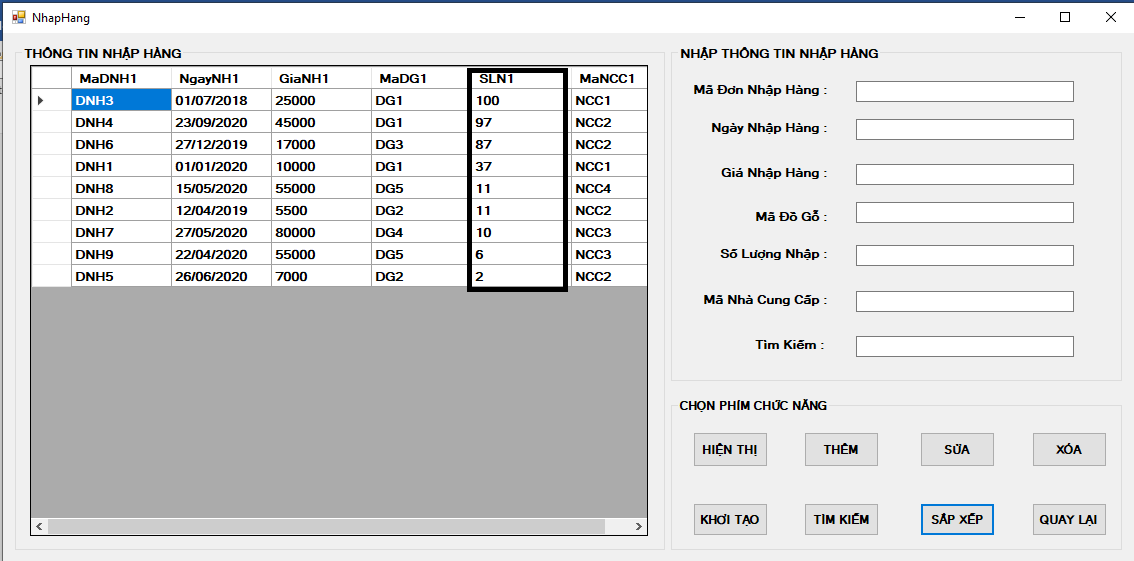
|  |
| --- |
| void HienThi1()  {  string sqlconn = @"Data Source=DESKTOP-Q6N4J46\SQLEXPRESS01;Initial Catalog=QLCuaHangDoGo;Integrated Security=True";  SqlConnection conn = new SqlConnection(sqlconn);  conn.Open();  string cmd = "SELECT \* FROM DonDatHang ";  SqlCommand scmd = new SqlCommand(cmd);  List<DonDatHangSX> listnv = new List<DonDatHangSX>();  DataTable data = new DataTable();  SqlDataAdapter da = new SqlDataAdapter(cmd, conn);  da.Fill(data);  foreach (DataRow items in data.Rows)  {  DonDatHangSX nhaphang = new DonDatHangSX(items);  listnv.Add(nhaphang);  }  conn.Close();  dgvDonDatHang.DataSource = QuickSort(listnv, 0, listnv.Count - 1);  } |

**Trước khi sắp xếp đơn nhập hàng theo sô lượng nhập**

****

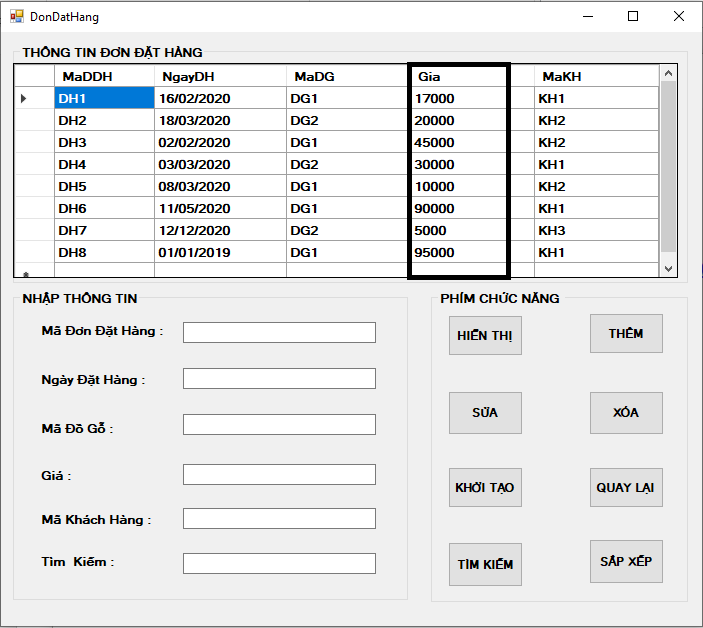
Hình Ảnh . Sắp xếp nhập hàng

**Sau khi click sắp xếp theo số lượng nhập**

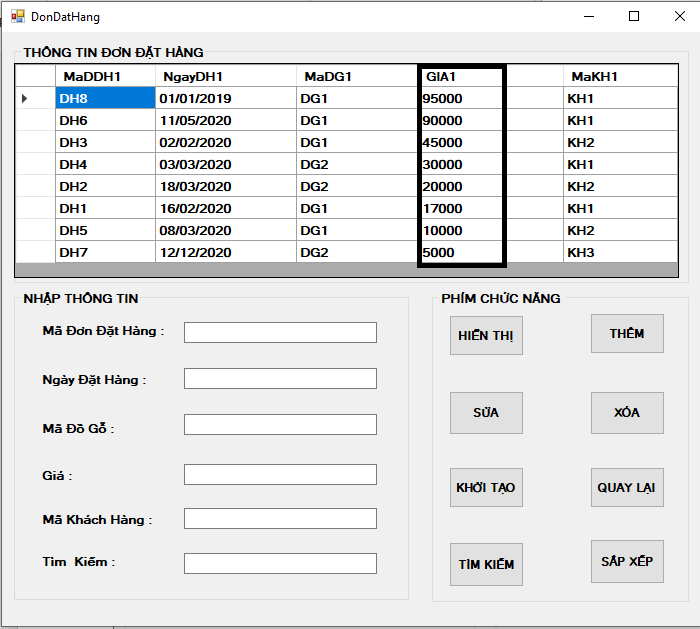
****

Hình Ảnh 24. Sắp xếp đơn nhập hàng theo số lượng nhập nhập hàng

**Trước khi sắp xếp đơn đặt hàng theo giá bán**

****

**Sau khi click sắp xếp đơn đặt hàng theo giá bán**

****

Hình Ảnh 25. Sắp xếp đơn đặt hàng theo giá bán

KẾT LUẬN

Nhờ có quicksort mà dữ liệu đơn hàng được sắp xếp dễ dàng hơn nên có nhiều lợi ích cho cửa hàng.

Hóa đơn là yếu tố quan trọng nhất trong quá trình hoạt động của bất kỳ một doanh nghiệp, cửa hàng. Do đó việc khai thác tốt nguồn lực này để phục vụ phát triển doanh nghiệp và xã hội là một vấn đề quan trọng trong việc quản lý các đơn hàng. Việc quản lý đòi hỏi sự hiểu biết về con người ở nhiều khía cạnh, và quan niệm rằng con người là yếu tố trung tâm của sự phát triển.

Đối với cửa hàng:

Cung cấp thông tin chính xác, kịp thời và toàn diện về tình đơn hàng, nhập hàng qua đó hỗ trợ công tác hoạch định và ra quyết định hiệu quả.

Có được công cụ hỗ trợ hữu ích để phục vụ công tác chuyên môn, lưu trữ dữ liệu khoa học, đánh giá đơn hàng chính xác, nhanh chóng nhằm giảm thời gian tránh lãng phí trong công việc.

Đối với nhân viên:

Có được thông tin chi tiết, rõ ràng. Dễ dàng theo dõi đối chiếu tra cứu thông tin.

TÀI LIỆU THAM KHẢO

[1]Đỗ Xuân Lôi; *Giáo trình cấu trúc dữ liệu và giải thuật*; NXB Đại học quốc gia Hà Nội

[2]Website: vi.wikipedia.org, stdio.vn

[3] Nguyễn Ngọc Cương; *Giáo trình cơ sở dữ liệu;* NXB Thông tin và truyền thông

[4]Lê Trung Hiếu *Giáo trình lập trình windows from và C#.Net;* NXB Thông tin và truyền