Welcome back! What would you like to chat about?

Received message. Welcome back! What would you like to chat about?

Tôi có 1 bảng gồm mã hàng, giá bán. tôi muốn tổng hợp lại thông tin mã hàng và giá bán, trong đó các mã hàng trùng nhau sẽ tổng lại và hiển thị 1 lần duy nhất trong bảng. tôi dùng c#\

Sent message. Tôi có 1 bảng gồm mã hàng, giá bán. tôi muốn tổng hợp lại thông tin mã hàng và giá bán, trong đó các mã hàng trùng nhau sẽ tổng lại và hiển thị 1 lần duy nhất trong bảng. tôi dùng c#\

Searching for: c# group by sum

Searching for: c# group by sum

Generating answers for you…

Generating answers for you…

Bạn có thể sử dụng Linq trong C# để nhóm các mã hàng và tính tổng giá bán của chúng. Đây là một ví dụ về cách bạn có thể làm điều đó:

var result = data.GroupBy(d => d.ProductCode)

.Select(g => new

{

ProductCode = g.Key,

TotalPrice = g.Sum(s => s.Price)

});

Copy

Trong đó data là danh sách các sản phẩm của bạn, mỗi sản phẩm có thuộc tính ProductCode và Price. Kết quả sẽ là một danh sách các đối tượng mới, mỗi đối tượng có thuộc tính ProductCode và TotalPrice, trong đó TotalPrice là tổng giá bán của tất cả các sản phẩm có cùng mã hàng[1](https://stackoverflow.com/questions/16522645/linq-groupby-sum-and-count). Bạn có thể tham khảo thêm tại [Stack Overflow](https://stackoverflow.com/questions/16522645/linq-groupby-sum-and-count) để biết thêm chi tiết.

Received message. Bạn có thể sử dụng Linq trong C# để nhóm các mã hàng và tính tổng giá bán của chúng. Đây là một ví dụ về cách bạn có thể làm điều đó: ```csharp var result = data.GroupBy(d => d.ProductCode) .Select(g => new { ProductCode = g.Key, TotalPrice = g.Sum(s => s.Price) }); ``` Trong đó `data` là danh sách các sản phẩm của bạn, mỗi sản phẩm có thuộc tính `ProductCode` và `Price`. Kết quả sẽ là một danh sách các đối tượng mới, mỗi đối tượng có thuộc tính `ProductCode` và `TotalPrice`, trong đó `TotalPrice` là tổng giá bán của tất cả các sản phẩm có cùng mã hàng. Bạn có thể tham khảo thêm tại [Stack Overflow](https://stackoverflow.com/questions/16522645/linq-groupby-sum-and-count) để biết thêm chi tiết.

Learn more:

[1. stackoverflow.com](https://stackoverflow.com/questions/16522645/linq-groupby-sum-and-count)[2. stackoverflow.com](https://stackoverflow.com/questions/16100900/select-multiple-fields-group-by-and-sum)[3. riptutorial.com](https://riptutorial.com/csharp/example/17012/groupby-sum-and-count)[4. stackoverflow.com](https://stackoverflow.com/questions/57596738/group-and-sum-datatable)+2 more

1of30

tôi đang dùng mảng để lưu trữ

Sent message. tôi đang dùng mảng để lưu trữ

Nếu bạn đang sử dụng mảng để lưu trữ dữ liệu thì bạn có thể chuyển đổi mảng sang danh sách trước khi sử dụng Linq. Đây là một ví dụ về cách bạn có thể làm điều đó:

var list = array.ToList();

var result = list.GroupBy(d => d.ProductCode)

.Select(g => new

{

ProductCode = g.Key,

TotalPrice = g.Sum(s => s.Price)

});

Copy

Trong đó array là mảng các sản phẩm của bạn. Bạn có thể sử dụng phương thức ToList() để chuyển đổi mảng sang danh sách và sau đó sử dụng Linq để nhóm và tính tổng giá bán như đã trình bày ở trên.

Received message. Nếu bạn đang sử dụng mảng để lưu trữ dữ liệu thì bạn có thể chuyển đổi mảng sang danh sách trước khi sử dụng Linq. Đây là một ví dụ về cách bạn có thể làm điều đó: ```csharp var list = array.ToList(); var result = list.GroupBy(d => d.ProductCode) .Select(g => new { ProductCode = g.Key, TotalPrice = g.Sum(s => s.Price) }); ``` Trong đó `array` là mảng các sản phẩm của bạn. Bạn có thể sử dụng phương thức `ToList()` để chuyển đổi mảng sang danh sách và sau đó sử dụng Linq để nhóm và tính tổng giá bán như đã trình bày ở trên.

2of30

tôi vẫn muốn dùng mảng

Sent message. tôi vẫn muốn dùng mảng

Nếu bạn muốn tiếp tục sử dụng mảng thì bạn vẫn có thể sử dụng Linq để nhóm và tính tổng giá bán. Đây là một ví dụ về cách bạn có thể làm điều đó:

var result = array.GroupBy(d => d.ProductCode)

.Select(g => new

{

ProductCode = g.Key,

TotalPrice = g.Sum(s => s.Price)

});

Copy

Trong đó array là mảng các sản phẩm của bạn. Bạn có thể sử dụng Linq trực tiếp trên mảng mà không cần chuyển đổi sang danh sách. Kết quả sẽ giống như đã trình bày ở trên.

Received message. Nếu bạn muốn tiếp tục sử dụng mảng thì bạn vẫn có thể sử dụng Linq để nhóm và tính tổng giá bán. Đây là một ví dụ về cách bạn có thể làm điều đó: ```csharp var result = array.GroupBy(d => d.ProductCode) .Select(g => new { ProductCode = g.Key, TotalPrice = g.Sum(s => s.Price) }); ``` Trong đó `array` là mảng các sản phẩm của bạn. Bạn có thể sử dụng Linq trực tiếp trên mảng mà không cần chuyển đổi sang danh sách. Kết quả sẽ giống như đã trình bày ở trên.

3of30

ví dụ dữ liệu của tôi có thêm số hóa đơn, ngày hóa đơn, mã hàng thì sao

Sent message. ví dụ dữ liệu của tôi có thêm số hóa đơn, ngày hóa đơn, mã hàng thì sao

Nếu dữ liệu của bạn có thêm các thuộc tính như số hóa đơn và ngày hóa đơn thì bạn có thể thêm chúng vào đối tượng kết quả khi nhóm và tính tổng giá bán. Đây là một ví dụ về cách bạn có thể làm điều đó:

var result = array.GroupBy(d => d.ProductCode)

.Select(g => new

{

ProductCode = g.Key,

TotalPrice = g.Sum(s => s.Price),

InvoiceNumbers = g.Select(s => s.InvoiceNumber).Distinct(),

InvoiceDates = g.Select(s => s.InvoiceDate).Distinct()

});

Copy

Trong đó array là mảng các sản phẩm của bạn. Kết quả sẽ là một danh sách các đối tượng mới, mỗi đối tượng có thuộc tính ProductCode, TotalPrice, InvoiceNumbers và InvoiceDates. Trong đó TotalPrice là tổng giá bán của tất cả các sản phẩm có cùng mã hàng, InvoiceNumbers là danh sách các số hóa đơn duy nhất và InvoiceDates là danh sách các ngày hóa đơn duy nhất.

Bạn có thể thay đổi cách nhóm và tính tổng để phù hợp với yêu cầu của mình.

Received message. Nếu dữ liệu của bạn có thêm các thuộc tính như số hóa đơn và ngày hóa đơn thì bạn có thể thêm chúng vào đối tượng kết quả khi nhóm và tính tổng giá bán. Đây là một ví dụ về cách bạn có thể làm điều đó: ```csharp var result = array.GroupBy(d => d.ProductCode) .Select(g => new { ProductCode = g.Key, TotalPrice = g.Sum(s => s.Price), InvoiceNumbers = g.Select(s => s.InvoiceNumber).Distinct(), InvoiceDates = g.Select(s => s.InvoiceDate).Distinct() }); ``` Trong đó `array` là mảng các sản phẩm của bạn. Kết quả sẽ là một danh sách các đối tượng mới, mỗi đối tượng có thuộc tính `ProductCode`, `TotalPrice`, `InvoiceNumbers` và `InvoiceDates`. Trong đó `TotalPrice` là tổng giá bán của tất cả các sản phẩm có cùng mã hàng, `InvoiceNumbers` là danh sách các số hóa đơn duy nhất và `InvoiceDates` là danh sách các ngày hóa đơn duy nhất. Bạn có thể thay đổi cách nhóm và tính tổng để phù hợp với yêu cầu của mình.

4of30

sau khi gộp xong thì tôi viết lại mảng mới như thế nào?

Sent message. sau khi gộp xong thì tôi viết lại mảng mới như thế nào?

Sau khi bạn đã nhóm và tính tổng giá bán xong thì bạn có thể chuyển đổi kết quả sang một mảng mới nếu muốn. Đây là một ví dụ về cách bạn có thể làm điều đó:

var result = array.GroupBy(d => d.ProductCode)

.Select(g => new

{

ProductCode = g.Key,

TotalPrice = g.Sum(s => s.Price),

InvoiceNumbers = g.Select(s => s.InvoiceNumber).Distinct(),

InvoiceDates = g.Select(s => s.InvoiceDate).Distinct()

})

.ToArray();

Copy

Trong đó array là mảng các sản phẩm của bạn. Bạn có thể sử dụng phương thức ToArray() để chuyển đổi kết quả sang một mảng mới. Mảng mới này sẽ chứa các đối tượng kết quả sau khi nhóm và tính tổng giá bán.

Received message. Sau khi bạn đã nhóm và tính tổng giá bán xong thì bạn có thể chuyển đổi kết quả sang một mảng mới nếu muốn. Đây là một ví dụ về cách bạn có thể làm điều đó: ```csharp var result = array.GroupBy(d => d.ProductCode) .Select(g => new { ProductCode = g.Key, TotalPrice = g.Sum(s => s.Price), InvoiceNumbers = g.Select(s => s.InvoiceNumber).Distinct(), InvoiceDates = g.Select(s => s.InvoiceDate).Distinct() }) .ToArray(); ``` Trong đó `array` là mảng các sản phẩm của bạn. Bạn có thể sử dụng phương thức `ToArray()` để chuyển đổi kết quả sang một mảng mới. Mảng mới này sẽ chứa các đối tượng kết quả sau khi nhóm và tính tổng giá bán.

5of30

cho tôi code cụ thể

Sent message. cho tôi code cụ thể

Dưới đây là một ví dụ cụ thể về cách bạn có thể nhóm và tính tổng giá bán của các sản phẩm trong một mảng và chuyển đổi kết quả sang một mảng mới:

// Định nghĩa lớp Product

public class Product

{

public string InvoiceNumber { get; set; }

public DateTime InvoiceDate { get; set; }

public string ProductCode { get; set; }

public decimal Price { get; set; }

}

// Khởi tạo mảng các sản phẩm

var array = new Product[]

{

new Product { InvoiceNumber = "INV001", InvoiceDate = new DateTime(2022, 1, 1), ProductCode = "P001", Price = 10 },

new Product { InvoiceNumber = "INV002", InvoiceDate = new DateTime(2022, 1, 2), ProductCode = "P001", Price = 20 },

new Product { InvoiceNumber = "INV003", InvoiceDate = new DateTime(2022, 1, 3), ProductCode = "P002", Price = 30 },

};

// Nhóm và tính tổng giá bán

var result = array.GroupBy(d => d.ProductCode)

.Select(g => new

{

ProductCode = g.Key,

TotalPrice = g.Sum(s => s.Price),

InvoiceNumbers = g.Select(s => s.InvoiceNumber).Distinct(),

InvoiceDates = g.Select(s => s.InvoiceDate).Distinct()

})

.ToArray();

// In kết quả ra màn hình

foreach (var item in result)

{

Console.WriteLine($"ProductCode: {item.ProductCode}, TotalPrice: {item.TotalPrice}");

Console.WriteLine($"InvoiceNumbers: {string.Join(", ", item.InvoiceNumbers)}");

Console.WriteLine($"InvoiceDates: {string.Join(", ", item.InvoiceDates)}");

}

Copy

Trong đó array là mảng các sản phẩm của bạn. Kết quả sẽ là một mảng mới chứa các đối tượng kết quả sau khi nhóm và tính tổng giá bán. Bạn có thể thay đổi cách nhóm và tính tổng để phù hợp với yêu cầu của mình.

Received message. Dưới đây là một ví dụ cụ thể về cách bạn có thể nhóm và tính tổng giá bán của các sản phẩm trong một mảng và chuyển đổi kết quả sang một mảng mới: ```csharp // Định nghĩa lớp Product public class Product { public string InvoiceNumber { get; set; } public DateTime InvoiceDate { get; set; } public string ProductCode { get; set; } public decimal Price { get; set; } } // Khởi tạo mảng các sản phẩm var array = new Product[] { new Product { InvoiceNumber = "INV001", InvoiceDate = new DateTime(2022, 1, 1), ProductCode = "P001", Price = 10 }, new Product { InvoiceNumber = "INV002", InvoiceDate = new DateTime(2022, 1, 2), ProductCode = "P001", Price = 20 }, new Product { InvoiceNumber = "INV003", InvoiceDate = new DateTime(2022, 1, 3), ProductCode = "P002", Price = 30 }, }; // Nhóm và tính tổng giá bán var result = array.GroupBy(d => d.ProductCode) .Select(g => new { ProductCode = g.Key, TotalPrice = g.Sum(s => s.Price), InvoiceNumbers = g.Select(s => s.InvoiceNumber).Distinct(), InvoiceDates = g.Select(s => s.InvoiceDate).Distinct() }) .ToArray(); // In kết quả ra màn hình foreach (var item in result) { Console.WriteLine($"ProductCode: {item.ProductCode}, TotalPrice: {item.TotalPrice}"); Console.WriteLine($"InvoiceNumbers: {string.Join(", ", item.InvoiceNumbers)}"); Console.WriteLine($"InvoiceDates: {string.Join(", ", item.InvoiceDates)}"); } ``` Trong đó `array` là mảng các sản phẩm của bạn. Kết quả sẽ là một mảng mới chứa các đối tượng kết quả sau khi nhóm và tính tổng giá bán. Bạn có thể thay đổi cách nhóm và tính tổng để phù hợp với yêu cầu của mình.

6of30

giúp tôi viết tiếp public static void LuuDuLieuHTK(QL\_CuaHang DuLieu) { int tongsoluong = 0; QL\_CuaHang[] \_dulieu = DocDuLieu(); QL\_CuaHang[] newDulieu = new QL\_CuaHang[\_dulieu.Length + 1]; //khoi tao mang voi kich thuoc lon hon 1 don vi so voi mang ban dau // Tinh toan gop ma QL\_CuaHang[] newDL = DocDuLieuXuathang(); QL\_CuaHang[] newArr = \_dulieu.Concat(newDL).ToArray(); QL\_CuaHang[] TH = newArr; //Gộp mã hàng var result = newArr.GroupBy(d => d.mahang).Select(g => new { mahang = g.Key, thanhtien = g.Sum(s => s.thanhtien), tenHH = g.Select(s => s.tenhang).Distinct(), soluongM = g.Sum(s => s.soluong), soluongB = g.Sum(s => s.soluongXuat), loaiHang = g.Select(s => s.loaihang).Distinct(), nXS = g.Select(s => s.nhasanxuat).Distinct() }).ToArray();

Sent message. giúp tôi viết tiếp public static void LuuDuLieuHTK(QL\_CuaHang DuLieu) { int tongsoluong = 0; QL\_CuaHang[] \_dulieu = DocDuLieu(); QL\_CuaHang[] newDulieu = new QL\_CuaHang[\_dulieu.Length + 1]; //khoi tao mang voi kich thuoc lon hon 1 don vi so voi mang ban dau // Tinh toan gop ma QL\_CuaHang[] newDL = DocDuLieuXuathang(); QL\_CuaHang[] newArr = \_dulieu.Concat(newDL).ToArray(); QL\_CuaHang[] TH = newArr; //Gộp mã hàng var result = newArr.GroupBy(d => d.mahang).Select(g => new { mahang = g.Key, thanhtien = g.Sum(s => s.thanhtien), tenHH = g.Select(s => s.tenhang).Distinct(), soluongM = g.Sum(s => s.soluong), soluongB = g.Sum(s => s.soluongXuat), loaiHang = g.Select(s => s.loaihang).Distinct(), nXS = g.Select(s => s.nhasanxuat).Distinct() }).ToArray();

Dưới đây là một ví dụ về cách bạn có thể tiếp tục viết phương thức LuuDuLieuHTK:

public static void LuuDuLieuHTK(QL\_CuaHang DuLieu)

{

int tongsoluong = 0;

QL\_CuaHang[] \_dulieu = DocDuLieu();

QL\_CuaHang[] newDulieu = new QL\_CuaHang[\_dulieu.Length + 1];

// Tinh toan gop ma

QL\_CuaHang[] newDL = DocDuLieuXuathang();

QL\_CuaHang[] newArr = \_dulieu.Concat(newDL).ToArray();

QL\_CuaHang[] TH = newArr;

//Gộp mã hàng

var result = newArr.GroupBy(d => d.mahang).Select(g => new

{

mahang = g.Key,

thanhtien = g.Sum(s => s.thanhtien),

tenHH = g.Select(s => s.tenhang).Distinct(),

soluongM = g.Sum(s => s.soluong),

soluongB = g.Sum(s => s.soluongXuat),

loaiHang = g.Select(s => s.loaihang).Distinct(),

nXS = g.Select(s => s.nhasanxuat).Distinct()

}).ToArray();

// Tiếp tục viết code ở đây

}

Copy

Trong đó newArr là mảng mới chứa dữ liệu sau khi đã gộp từ hai mảng \_dulieu và newDL. Kết quả sẽ là một mảng mới chứa các đối tượng kết quả sau khi nhóm và tính tổng giá bán.

Bạn có thể tiếp tục viết code để xử lý kết quả như mong muốn. Nếu bạn cần thêm trợ giúp thì hãy cho tôi biết yêu cầu cụ thể của bạn.

Received message. Dưới đây là một ví dụ về cách bạn có thể tiếp tục viết phương thức `LuuDuLieuHTK`: ```csharp public static void LuuDuLieuHTK(QL\_CuaHang DuLieu) { int tongsoluong = 0; QL\_CuaHang[] \_dulieu = DocDuLieu(); QL\_CuaHang[] newDulieu = new QL\_CuaHang[\_dulieu.Length + 1]; // Tinh toan gop ma QL\_CuaHang[] newDL = DocDuLieuXuathang(); QL\_CuaHang[] newArr = \_dulieu.Concat(newDL).ToArray(); QL\_CuaHang[] TH = newArr; //Gộp mã hàng var result = newArr.GroupBy(d => d.mahang).Select(g => new { mahang = g.Key, thanhtien = g.Sum(s => s.thanhtien), tenHH = g.Select(s => s.tenhang).Distinct(), soluongM = g.Sum(s => s.soluong), soluongB = g.Sum(s => s.soluongXuat), loaiHang = g.Select(s => s.loaihang).Distinct(), nXS = g.Select(s => s.nhasanxuat).Distinct() }).ToArray(); // Tiếp tục viết code ở đây } ``` Trong đó `newArr` là mảng mới chứa dữ liệu sau khi đã gộp từ hai mảng `\_dulieu` và `newDL`. Kết quả sẽ là một mảng mới chứa các đối tượng kết quả sau khi nhóm và tính tổng giá bán. Bạn có thể tiếp tục viết code để xử lý kết quả như mong muốn. Nếu bạn cần thêm trợ giúp thì hãy cho tôi biết yêu cầu cụ thể của bạn.

7of30

viết tiếp mảng mới

Sent message. viết tiếp mảng mới

Sau khi bạn đã nhóm và tính tổng giá bán xong thì bạn có thể chuyển đổi kết quả sang một mảng mới nếu muốn. Đây là một ví dụ về cách bạn có thể làm điều đó trong phương thức LuuDuLieuHTK:

public static void LuuDuLieuHTK(QL\_CuaHang DuLieu)

{

int tongsoluong = 0;

QL\_CuaHang[] \_dulieu = DocDuLieu();

QL\_CuaHang[] newDulieu = new QL\_CuaHang[\_dulieu.Length + 1];

// Tinh toan gop ma

QL\_CuaHang[] newDL = DocDuLieuXuathang();

QL\_CuaHang[] newArr = \_dulieu.Concat(newDL).ToArray();

QL\_CuaHang[] TH = newArr;

//Gộp mã hàng

var result = newArr.GroupBy(d => d.mahang).Select(g => new

{

mahang = g.Key,

thanhtien = g.Sum(s => s.thanhtien),

tenHH = g.Select(s => s.tenhang).Distinct(),

soluongM = g.Sum(s => s.soluong),

soluongB = g.Sum(s => s.soluongXuat),

loaiHang = g.Select(s => s.loaihang).Distinct(),

nXS = g.Select(s => s.nhasanxuat).Distinct()

}).ToArray();

// Chuyển đổi kết quả sang một mảng mới

var newArray = result.Select(r => new QL\_CuaHang

{

mahang = r.mahang,

thanhtien = r.thanhtien,

tenhang = r.tenHH.FirstOrDefault(),

soluong = r.soluongM,

soluongXuat = r.soluongB,

loaihang = r.loaiHang.FirstOrDefault(),

nhasanxuat = r.nXS.FirstOrDefault()

}).ToArray();

// Tiếp tục viết code ở đây

}

Copy

Trong đó result là mảng chứa các đối tượng kết quả sau khi nhóm và tính tổng giá bán. Bạn có thể sử dụng phương thức Select() để chuyển đổi các đối tượng kết quả sang các đối tượng QL\_CuaHang mới và sau đó sử dụng phương thức ToArray() để chuyển đổi sang một mảng mới.

Mảng mới này sẽ chứa các đối tượng QL\_CuaHang mới với các thuộc tính được cập nhật từ kết quả nhóm và tính tổng giá bán. Bạn có thể tiếp tục viết code để xử lý mảng mới này như mong muốn.

Received message. Sau khi bạn đã nhóm và tính tổng giá bán xong thì bạn có thể chuyển đổi kết quả sang một mảng mới nếu muốn. Đây là một ví dụ về cách bạn có thể làm điều đó trong phương thức `LuuDuLieuHTK`: ```csharp public static void LuuDuLieuHTK(QL\_CuaHang DuLieu) { int tongsoluong = 0; QL\_CuaHang[] \_dulieu = DocDuLieu(); QL\_CuaHang[] newDulieu = new QL\_CuaHang[\_dulieu.Length + 1]; // Tinh toan gop ma QL\_CuaHang[] newDL = DocDuLieuXuathang(); QL\_CuaHang[] newArr = \_dulieu.Concat(newDL).ToArray(); QL\_CuaHang[] TH = newArr; //Gộp mã hàng var result = newArr.GroupBy(d => d.mahang).Select(g => new { mahang = g.Key, thanhtien = g.Sum(s => s.thanhtien), tenHH = g.Select(s => s.tenhang).Distinct(), soluongM = g.Sum(s => s.soluong), soluongB = g.Sum(s => s.soluongXuat), loaiHang = g.Select(s => s.loaihang).Distinct(), nXS = g.Select(s => s.nhasanxuat).Distinct() }).ToArray(); // Chuyển đổi kết quả sang một mảng mới var newArray = result.Select(r => new QL\_CuaHang { mahang = r.mahang, thanhtien = r.thanhtien, tenhang = r.tenHH.FirstOrDefault(), soluong = r.soluongM, soluongXuat = r.soluongB, loaihang = r.loaiHang.FirstOrDefault(), nhasanxuat = r.nXS.FirstOrDefault() }).ToArray(); // Tiếp tục viết code ở đây } ``` Trong đó `result` là mảng chứa các đối tượng kết quả sau khi nhóm và tính tổng giá bán. Bạn có thể sử dụng phương thức `Select()` để chuyển đổi các đối tượng kết quả sang các đối tượng `QL\_CuaHang` mới và sau đó sử dụng phương thức `ToArray()` để chuyển đổi sang một mảng mới. Mảng mới này sẽ chứa các đối tượng `QL\_CuaHang` mới với các thuộc tính được cập nhật từ kết quả nhóm và tính tổng giá bán. Bạn có thể tiếp tục viết code để xử lý mảng mới này như mong muốn.

8of30

ngoài linq ra không còn cách nào đơn giản hơn không?

Sent message. ngoài linq ra không còn cách nào đơn giản hơn không?

Ngoài việc sử dụng Linq để nhóm và tính tổng giá bán thì bạn cũng có thể sử dụng các vòng lặp và điều kiện để thực hiện điều đó. Tuy nhiên, việc sử dụng Linq sẽ giúp cho code của bạn ngắn gọn và dễ đọc hơn.

Dưới đây là một ví dụ về cách bạn có thể nhóm và tính tổng giá bán mà không sử dụng Linq:

var result = new List<QL\_CuaHang>();

foreach (var item in newArr)

{

var existingItem = result.FirstOrDefault(r => r.mahang == item.mahang);

if (existingItem != null)

{

existingItem.thanhtien += item.thanhtien;

existingItem.soluong += item.soluong;

existingItem.soluongXuat += item.soluongXuat;

}

else

{

result.Add(new QL\_CuaHang

{

mahang = item.mahang,

thanhtien = item.thanhtien,

tenhang = item.tenhang,

soluong = item.soluong,

soluongXuat = item.soluongXuat,

loaihang = item.loaihang,

nhasanxuat = item.nhasanxuat

});

}

}

Copy

Trong đó newArr là mảng mới chứa dữ liệu sau khi đã gộp từ hai mảng \_dulieu và newDL. Kết quả sẽ là một danh sách mới chứa các đối tượng kết quả sau khi nhóm và tính tổng giá bán.

Bạn có thể thay đổi cách nhóm và tính tổng để phù hợp với yêu cầu của mình. Tuy nhiên, tôi khuyên bạn nên sử dụng Linq để code của bạn ngắn gọn và dễ đọc hơn.

Received message. Ngoài việc sử dụng Linq để nhóm và tính tổng giá bán thì bạn cũng có thể sử dụng các vòng lặp và điều kiện để thực hiện điều đó. Tuy nhiên, việc sử dụng Linq sẽ giúp cho code của bạn ngắn gọn và dễ đọc hơn. Dưới đây là một ví dụ về cách bạn có thể nhóm và tính tổng giá bán mà không sử dụng Linq: ```csharp var result = new List<QL\_CuaHang>(); foreach (var item in newArr) { var existingItem = result.FirstOrDefault(r => r.mahang == item.mahang); if (existingItem != null) { existingItem.thanhtien += item.thanhtien; existingItem.soluong += item.soluong; existingItem.soluongXuat += item.soluongXuat; } else { result.Add(new QL\_CuaHang { mahang = item.mahang, thanhtien = item.thanhtien, tenhang = item.tenhang, soluong = item.soluong, soluongXuat = item.soluongXuat, loaihang = item.loaihang, nhasanxuat = item.nhasanxuat }); } } ``` Trong đó `newArr` là mảng mới chứa dữ liệu sau khi đã gộp từ hai mảng `\_dulieu` và `newDL`. Kết quả sẽ là một danh sách mới chứa các đối tượng kết quả sau khi nhóm và tính tổng giá bán. Bạn có thể thay đổi cách nhóm và tính tổng để phù hợp với yêu cầu của mình. Tuy nhiên, tôi khuyên bạn nên sử dụng Linq để code của bạn ngắn gọn và dễ đọc hơn.

9of30