**HOMEWORK** : 9/3/2025

**NAME** : Võ Thị Thúy Sang - 22IT248

**CLASS** : Lập trình web nâng cao (1)

* delegate : cơ chế ủy quyền - cho phép thực thi 1 phương thức nào đó
* single: 1 lần nhận 1 phương thức
* multicast delegate: dữ liệu trả về kiểu void, nhận nhiều phương thức
* fun: nhiều tham số đầu vào - 1 đầu ra (có giá trị trả về)
* action: return luôn về kiểu void, không có tham số đầu ra
* khai báo delegate giống khai báo một phương thức nhưng không có phần thân phương thức, thêm từ delegate

log = Info;

log("Xin chao"); hoặc log.Invoke("Xin chao ABC");

* biến delegate có thể 1 lúc tham chiếu đến nhiều phương thức - tạo ra một chuỗi các tham chiếu
* Toán tử: += Nối thêm một phương thức vào delegate, ví dụ delegatevar += method1
* Toán tử: -= Loại bỏ 1 phương ở cuối (nếu phương thức đó có trong delegate, tính từ cuối)
* ví dụ delegatevar -= method1
* Func là mẫu delegate có kiểu trả về. Để khai báo biến delegate dùng cú pháp như sau:

Func<kiểu\_tham\_số\_1, kiểu\_tham\_số\_2, ..., kiểu\_trả\_về> var\_delegate;

* Action tương tự như Func, điều khác duy nhất là nó không có kiểu trả về, khai báo cơ bản như sau:

Action<kiểu\_tham\_số\_1, kiểu\_tham\_số\_2, ... > var\_delegate;

* Lambda - phương thức vô danh - biểu thức có tham số đầu vào, thi hành các chỉ thị thính toán, có thể gán cho biến kiểu delegate
* Event là cơ chế để một đối tượng (đối tượng của lớp) này thông báo đến đối tượng khác có điều gì đó mà lớp khác quan tâm vừa xảy ra
* Lớp mà từ đó gửi đi sự kiện gọi tên nó là publisher và các lớp nhận được sự kiện gọi là là các subsriber.
* Linq: ngôn ngữ truy vấn tích hợp - tạo ra nguồn dữ liệu là một danh sách
* Câu truy vấn Linq thường bắt đầu bằng mệnh đề from và kết thúc bằng mệnh đề select hoặc group, giữa chúng là những mệnh đề where, orderby, join, let.
* Các phương thức:

/// Select  
/// SelectMany  
/// Where  
/// Min, Max, Sum, Average  
/// Join  
/// Group Join  
/// Take  
/// Skip  
/// OrderBy - OrderByDescending  
/// Reverse  
/// GroupBy  
/// Distinct  
/// Single - SingleOrDefault  
/// Any

///All  
/// Count ()

Bài 1: Sử dụng delegate trong C# hàm ủy quyền:

1. using System;
2. namespace CS\_Delagate
3. {
4. public delegate void ShowLog(string message);
5. class Program
6. {
7. static void Info(string s)
8. {
9. Console.ForegroundColor = ConsoleColor.Green;
10. Console.WriteLine(s);
11. Console.ResetColor();
12. }
13. static void Warning(string s)
14. {
15. Console.ForegroundColor = ConsoleColor.Red;
16. Console.WriteLine(s);
17. Console.ResetColor();
18. }
19. // static int Tong(int a, int b) => a + b;
20. //static int Hieu(int a, int b) => a - b;
21. static void Tong(int a, int b, ShowLog log)
22. {
23. int kq = a + b;
24. log?.Invoke($"Tong la {kq}");
25. }
26. static void Main(string[] args)
27. {
28. // Action action; // ~ delegate void KIEU();
29. // Action<string, int> action1;
30. // Action<string> action2;
31. // action2 = Warning;
32. // action2 += Info;
33. // action2?.Invoke("Thong bao tu Action");
34. // Func<int> f1; // ~ delegate int KIEU();
35. // Func<string, double, string> f2; // ~ delegate string KIEU(string s, double s);
36. // Func<int, int, int> tinhtoan; //~delegate int KIEU(inta, int b);
37. // int a = 5;
38. //int b = 10;
39. //tinhtoan = Tong;
40. //Console.WriteLine($"Tong la {tinhtoan(a, b)}");
41. Tong(4, 5, Info);
42. }
43. }
44. }

A screenshot of a computer

AI-generated content may be incorrect.

Bài 2: Biểu thức lambda trong C# sử dụng lambda với delegate:

using System;

using System.Linq;

namespace CS\_Lambda

{

/\*

Lambda - anonymous function

1)

(tham so) =>bieu\_thuc;

2)

(tham\_so) => {

}

\*/

class Program

{

static void Main(string[] args)

{

// Action<string> thongbao;

//thongbao = (string s) => Console.WriteLine(s); // ~delegate void KIEU(string s) = Action<string>

//for (int i = 0; i < 10; i++)

// {

// thongbao?.Invoke("xin chao");

// }

// (int a, int b) =>

// {

// int kq = a + b;

// return kq;

// }

// Action thongbao;

// thongbao = () => Console.WriteLine("Xin chao cac ban");

// thongbao?.Invoke();

//Action<string, string> welcome;

//welcome = (mgs, name) => Console.WriteLine(mgs + " " + name);

//welcome?.Invoke("Xin chao", "Thuy Sang");

//welcome?.Invoke("Chao mung", "Thsan");

//Action<string, string> welcome;

//welcome = (mgs, name) =>

//{

// Console.ForegroundColor = ConsoleColor.Yellow;

// Console.WriteLine(mgs + " " + name);

// Console.ResetColor();

// };

//welcome?.Invoke("Xin chao", "Thuy Sang");

//welcome?.Invoke("Chao mung", "Thsan");

//Func<int, int, int> tinhtoan;

//tinhtoan = (int a, int b) =>

//{

// int kq = a + b;

// return kq;

//};

//Console.WriteLine(tinhtoan.Invoke(5, 6));

//int[] mang = { 2, 4, 64, 5, 7, 8, 9, 33, 55 };

//var kq = mang.Select(

// (int x) =>

//{

// return Math.Sqrt(x);

//});

//foreach (var result in kq)

//{

// Console.WriteLine(result);

// }

//int[] mang = { 2, 4, 64, 5, 7, 8, 9, 33, 55 };

//mang.ToList().ForEach(

// (int x) =>

// {

// if ( x % 2 != 0)

// Console.WriteLine(x);

// });

//int[] mang = { 2, 4, 64, 5, 7, 8, 9, 33, 55 };

//var kq = mang.Where(

// (x) =>

// {

// return x % 4 == 0;

// }

//

// );

//foreach (var n in kq)

//{

// Console.WriteLine(n);

//}

int[] mang = { 2, 4, 64, 5, 7, 8, 9, 33, 55 };

var kq = mang.Where(x => x >= 2 && x<= 35 );

foreach (var n in kq)

{

Console.WriteLine(n);

}

}

}

}

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

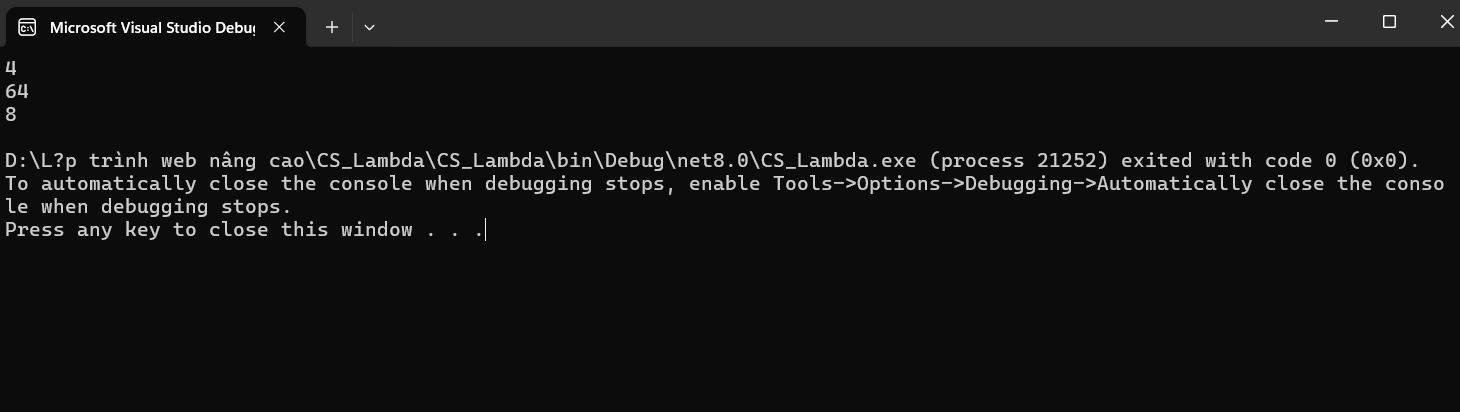
AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

Bài 3: Event trong C# các Event của .Net:

using System;

namespace CS\_Event

{

// publisher

public delegate void SuKienNhapSo(int x);

class Dulieunhap : EventArgs

{

public int data { get; set; }

public Dulieunhap(int x) => data = x;

}

class UserInput

{

//public event SuKienNhapSo sukiennhapso;

public event EventHandler sukiennhapso; // ~delegate void KIEU(object? sender, EventArgs args)

public void Input()

{

do

{

Console.Write("Nhap vao so nguyen: ");

string s = Console.ReadLine();

int i = Int32.Parse(s);

sukiennhapso?.Invoke(this, new Dulieunhap(i));

}

while (true);

}

}

class TinhCan

{

public void Sub(UserInput input)

{

input.sukiennhapso += Can;

}

public void Can(object sender, EventArgs e )

{

Dulieunhap dulieunhap = (Dulieunhap)e;

int i = dulieunhap.data;

Console.WriteLine($"Can bac hai cua {i} la {Math.Sqrt(i)}");

}

}

class BinhPhuong

{

public void Sub(UserInput input)

{

input.sukiennhapso += TinhBinhPhuong;

}

public void TinhBinhPhuong(object sender, EventArgs e)

{

Dulieunhap dulieunhap = (Dulieunhap)e;

int i = dulieunhap.data;

Console.WriteLine($"Binh phuong cua {i} la { i \* i }");

}

}

class Program

{

static void Main(string[] args)

{

Console.CancelKeyPress += (sender, e) =>

{

Console.WriteLine();

Console.WriteLine("Thoat ung dung");

};

//publisher

UserInput userInput = new UserInput();

userInput.sukiennhapso += (sender, e) =>

{

Dulieunhap dulieunhap = (Dulieunhap)e;

Console.WriteLine("Ban vua nhap so:"+ dulieunhap.data);

};

//subcriber

TinhCan tinhcan = new TinhCan();

tinhcan.Sub(userInput);

BinhPhuong binhPhuong = new BinhPhuong();

binhPhuong.Sub(userInput);

userInput.Input();

}

}

}

A black screen with a black border

AI-generated content may be incorrect.

A black screen with a black border

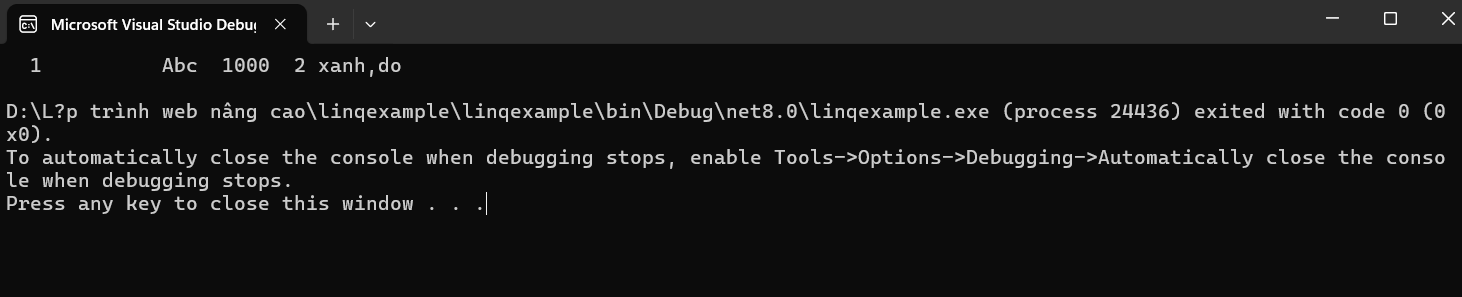
AI-generated content may be incorrect.

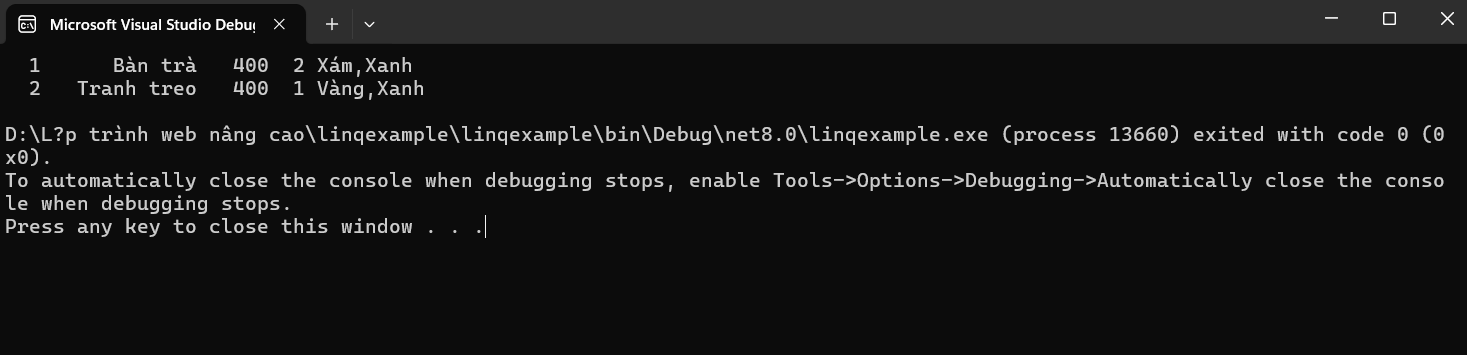
A black screen with a black border

AI-generated content may be incorrect.

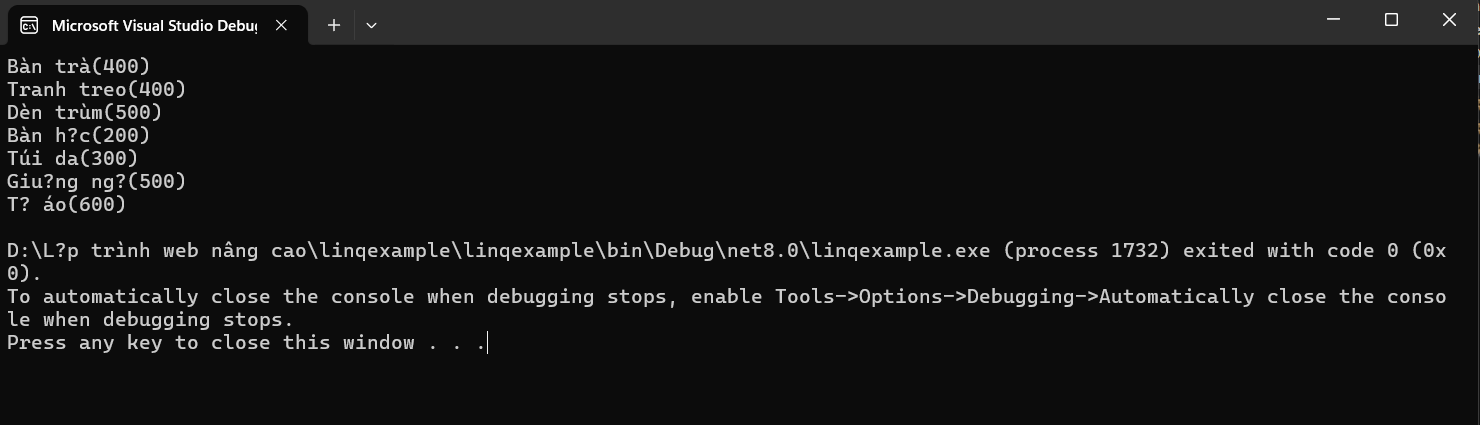
Bài 4: Linq trong lập trình C# .NET - thực hình ví dụ Linq:

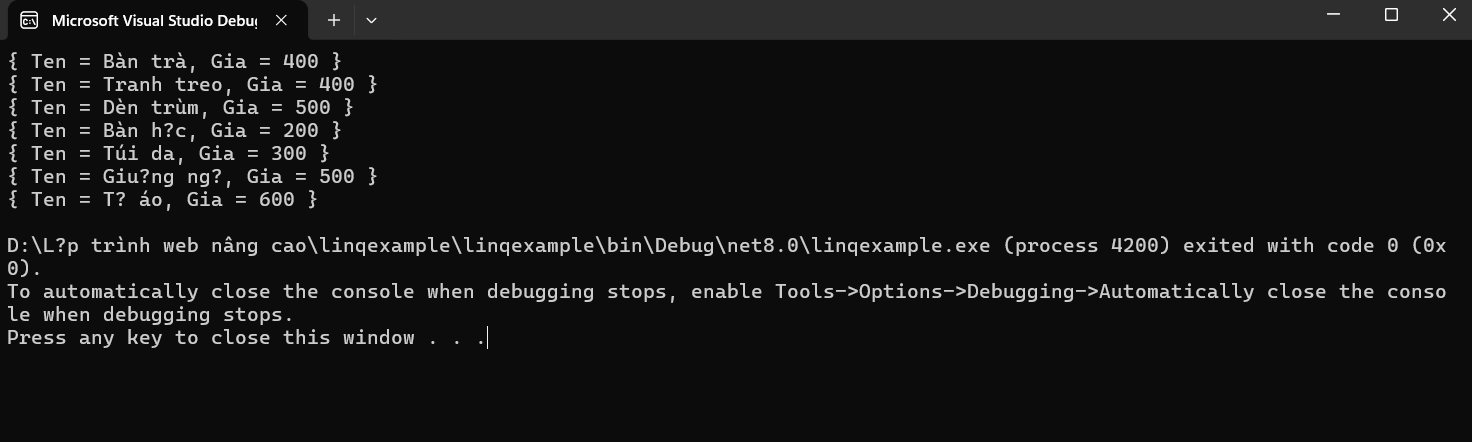
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. namespace linqexample
5. {
6. public class Product
7. {
8. public int ID { set; get; }
9. public string Name { set; get; } // tên
10. public double Price { set; get; } // giá
11. public string[] Colors { set; get; } // các màu sắc
12. public int Brand { set; get; } // ID Nhãn hiệu, hãng
13. public Product(int id, string name, double price, string[] colors, int brand)
14. {
15. ID = id; Name = name; Price = price; Colors = colors; Brand = brand;
16. }
17. // Lấy chuỗi thông tin sản phẳm gồm ID, Name, Price
18. override public string ToString()
19. => $"{ID,3} {Name,12} {Price,5} {Brand,2} {string.Join(",", Colors)}";
20. }
21. public class Brand
22. {
23. public string Name { set; get; }
24. public int ID { set; get; }
25. }
26. class Program
27. {
28. static void Main(string[] args)
29. {
30. //Product p = new Product(1, "Abc", 1000, new string[] {"xanh", "do"}, 2);
31. //Console.WriteLine(p);
32. var brands = new List<Brand>() {
33. new Brand{ID = 1, Name = "Công ty AAA"},
34. new Brand{ID = 2, Name = "Công ty BBB"},
35. new Brand{ID = 4, Name = "Công ty CCC"},
36. };
37. var products = new List<Product>()
38. {
39. new Product(1, "Bàn trà", 400, new string[] {"Xám", "Xanh"}, 2),
40. new Product(2, "Tranh treo", 400, new string[] {"Vàng", "Xanh"}, 1),
41. new Product(3, "Đèn trùm", 500, new string[] {"Trắng"}, 3),
42. new Product(4, "Bàn học", 200, new string[] {"Trắng", "Xanh"}, 1),
43. new Product(5, "Túi da", 300, new string[] {"Đỏ", "Đen", "Vàng"}, 2),
44. new Product(6, "Giường ngủ", 500, new string[] {"Trắng"}, 2),
45. new Product(7, "Tủ áo", 600, new string[] {"Trắng"}, 3),
46. };
47. int[] numbers = { 1, 2, 4, 6, 4, 2, 8, 9 };
48. /\*
49. 1) xac dinh nguon du lieu: from tenphantu in IEnumerables
50. ...join,where, oderby, let tenbien = ???...
51. 2) Lay du lieu: select, group by ...
52. \*/
53. //lay ra ten san pham
54. //var qr = from a in products select a.Name;
55. //foreach (var name in qr)
56. //{
57. // Console.WriteLine(name);
58. //}
59. //lay ra ten va gia san pham
60. //var qr = from a in products select $"{a.Name} ; {a.Price}";
61. //foreach (var name in qr)
62. //{
63. // Console.WriteLine(name);
64. //}
65. //lay ra san pham co gia bang 400
66. //var qr = from product in products
67. // where product.Price == 400
68. // select new
69. // {
70. // Ten = product.Name,
71. // Gia = product.Price,
72. // };
73. //foreach (var name in qr)
74. //{
75. // Console.WriteLine(name);
76. //}
77. //lay ra san pham co gia <= 500, mau Xanh
78. //var qr = from product in products
79. // from color in product.Colors
80. // where product.Price <= 500 && color == "Xanh"
81. // select new
82. // {
83. // Ten = product.Name,
84. // Gia = product.Price,
85. // Cacmau = product.Colors
86. // };
87. //foreach (var abc in qr)
88. //{
89. // Console.WriteLine($"{abc.Ten} - {abc.Gia}");
90. // Console.WriteLine(string.Join(',', abc.Cacmau));
91. //}
92. //nhom cac san pham theo gia
93. //var qr = from p in products
94. // group p by p.Price into gr
95. // orderby gr.Key
96. // select gr;
97. //qr.ToList().ForEach(group =>
98. //{
99. // Console.WriteLine(group.Key);
100. // group.ToList().ForEach(p => Console.WriteLine(p));
101. //});
102. // Doi tuong co:
103. // Gia
104. // Cacsanpham
105. // Soluong
106. //var qr = from p in products
107. // group p by p.Price into gr
108. // orderby gr.Key //tang dan ve gia
109. // let sl = "So luong la: " + gr.Count()
110. // select new
111. // {
112. // Gia = gr.Key,
113. // Cacsanpham = gr.ToList(),
114. // Soluong = sl
115. // };
116. //qr.ToList().ForEach(i =>
117. //{
118. // Console.WriteLine(i.Gia);
119. // Console.WriteLine(i.Soluong);
120. // i.Cacsanpham.ForEach(p => Console.WriteLine(p));
121. //});
122. //ket hop cac nguon du lieu
123. // liet ke cac san pham, ten, hang san xuat, gia cua san pham
124. var qr = from product in products
125. join brand in brands on product.Brand equals brand.ID into t
126. from b in t.DefaultIfEmpty()
127. select new
128. {
129. ten = product.Name,
130. gia = product.Price,
131. thuonghieu = (b != null) ? b.Name : "No Brand"
133. };
134. qr.ToList().ForEach(o =>
135. {
136. Console.WriteLine($"{o.ten,10} {o.thuonghieu,15} {o.gia,5}");
137. });
138. //Cac phuong thuc
139. // Select
140. // lay san pham co gia 400
141. //var query = from p in products where p.Price == 400 select p;
142. //foreach (var product in query)
143. //{
144. // Console.WriteLine(product);
145. //}
146. //lay danh sach ten cac san pham
147. //var kq = products.Select(
148. // (p) =>
149. // {
150. // return p.Name;
151. // }
152. // );
153. //foreach (var item in kq)
154. //{
155. // Console.WriteLine(item);
156. //}
157. //lay danh sach ten va gia san pham
158. //ar kq = products.Select(
159. // (p) =>
160. // {
161. // return p.Name + "(" + p.Price + ")";
162. // }
163. // );
164. //foreach (var item in kq)
165. //{
166. // Console.WriteLine(item);
167. //}
168. //tra ve du kieu kieu vo danh
169. //var kq = products.Select(
170. // (p) =>
171. // {
172. // return new {
173. // Ten = p.Name,
174. // Gia = p.Price,
175. // };
176. // }
177. // );
178. // foreach (var item in kq)
179. // {
180. // Console.WriteLine(item);
181. // }
182. // Where
183. //lay san pham ten co tr
184. //var kq = products.Where(
185. // (p) =>
186. // {
187. // return p.Name.Contains("tr");
188. // }
189. // );
190. //foreach (var item in kq)
191. //{
192. // Console.WriteLine(item);
193. //}
194. //lay san pham co gia >=200 va <=400
195. //var kq = products.Where(
196. // (p) =>
197. // {
198. // return p.Price >= 200 && p.Price <= 300;
199. // }
200. // );
201. //foreach (var item in kq)
202. //{
203. // Console.WriteLine(item);
204. //}
205. // SelectMany
206. //var kq = products.SelectMany(
207. // (p) =>
208. // {
209. // return p.Colors;
210. // }
211. // );
212. //foreach (var item in kq)
213. //{
214. // Console.WriteLine(item);
215. //}
216. // Min, Max, Sum, Average
217. //Console.WriteLine(numbers.Max());
218. //Console.WriteLine(numbers.Min());
219. //Console.WriteLine(numbers.Sum());
220. //Console.WriteLine(numbers.Average());
221. //so chan lon nhat
222. //Console.WriteLine(numbers.Where(n => n%2 == 0).Max());
223. //gia nho nhat cua san pham
224. //Console.WriteLine(products.Min(p => p.Price));
225. // Join - ket hop 2 nguon du lieu
226. //var kq = products.Join(brands, p => p.Brand, b => b.ID, (p, b) =>
227. //{
228. // return new
229. // {
230. // Ten = p.Name,
231. // Thuonghieu = b.Name,
232. // };
233. //});
234. // foreach (var item in kq)
235. //{
236. // Console.WriteLine(item);
237. //}
238. // GroupJoin
239. //var kq = brands.GroupJoin(products, b => b.ID, p => p.Brand,
240. // (brand, pros) =>
241. // {
242. // return new
243. // {
244. // Thuonghieu = brand.Name,
245. // Cacsanpham = pros,
246. // };
247. // }
248. //);
249. //foreach ( var gr in kq )
250. //{
251. // Console.WriteLine(gr.Thuonghieu);
252. // foreach(var p in gr.Cacsanpham)
253. // {
254. // Console.WriteLine(p);
255. // }
256. //}
257. // Take
258. // in thong tin 4 san pham dau tien
259. //products.Take(4).ToList().ForEach( p => Console.WriteLine(p));
260. // Skip
261. //bo qua thong tin 2 phan tu dau tien, lay nhung phan tu con lai
262. //products.Skip(2).ToList().ForEach(p => Console.WriteLine(p));
263. // OderBy (tang dan) - OderByDescending (giam dan)
264. //in san pham tang dan ve gia
265. //products.OrderBy(p => p.Price).ToList().ForEach(p => Console.WriteLine(p));
266. //in san pham giam dan ve gia
267. //products.OrderByDescending(p => p.Price).ToList().ForEach(p => Console.WriteLine(p));
268. // Reverse - dao nguoc thu tu cac phan tu trong tap hop
269. //numbers.Reverse().ToList().ForEach(n => Console.WriteLine(n));
270. // GroupBy
271. //nhom cac san pham theo gia
272. //var kq = products.GroupBy(p => p.Price);
273. //foreach(var group in kq)
274. //{
275. // Console.WriteLine(group.Key);
276. // foreach (var p in group)
277. // {
278. // Console.WriteLine(p);
279. // }
280. //}
282. // Distinct - loai bo nhung phan tu cung gia tri
283. //products.SelectMany(p => p.Colors).Distinct().ToList()
284. // .ForEach(mau => Console.WriteLine(mau));
286. // Single - kiem tra cac phan tu thoa man 1 dieu kien logic nao do - neu khong co se phat sinh loi
287. //var p = products.Single(p => p.Price == 600);
288. //Console.WriteLine(p);
290. //neu khong muon phai phat sinh loi - SingleOrDefault - khi khong tim thay tra ve kq la null
291. //var p = products.SingleOrDefault(p => p.Price == 1000);
292. //Console.WriteLine(p);
294. // Any - tra ve true neu thoa man 1 dieu kien logic nao do
295. //var p = products.Any(p => p.Price == 600);
296. //Console.WriteLine(p);
298. // All - tat ca cac phan tu phai thoa man 1 dieu kien logic nao do
299. //kiem tra cac phan tu deu co gia tren 200
300. //var p = products.All(p => p.Price >= 300);
301. //Console.WriteLine(p);
303. // Count - dem so luong
304. // dem tat ca cac san pham
305. //var p = products.Count();
306. //Console.WriteLine(p);
307. // dem tat ca cac san pham co gia tren 300
308. //var p = products.Count(p => p.Price >= 300);
309. //Console.WriteLine(p);
310. // In ra ten san pham, ten thuong hieu cos gia tu 300 - 400
311. // gia giam dan
312. //products.Where(p => p.Price >= 300 && p.Price <= 400)
313. // .OrderByDescending(p => p.Price)
314. // .Join(brands, p => p.Brand, b => b.ID, (sp, th) =>
315. // {
316. // return new
317. // {
318. // TenSP = sp.Name,
319. // TenTH = th.Name,
320. // Gia = sp.Price,
321. // };
322. // })
323. // .ToList()
324. // .ForEach(info =>
325. // {
326. // Console.WriteLine($"{info.TenSP, 15} {info.TenTH, 15}, {info.Gia, 5} ");
327. // });
328. }
329. }
330. }





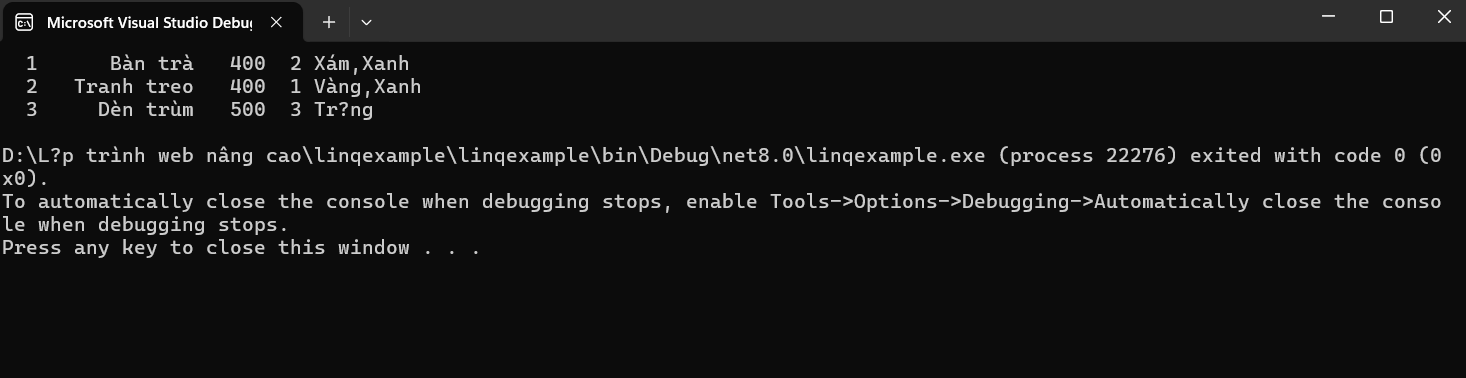


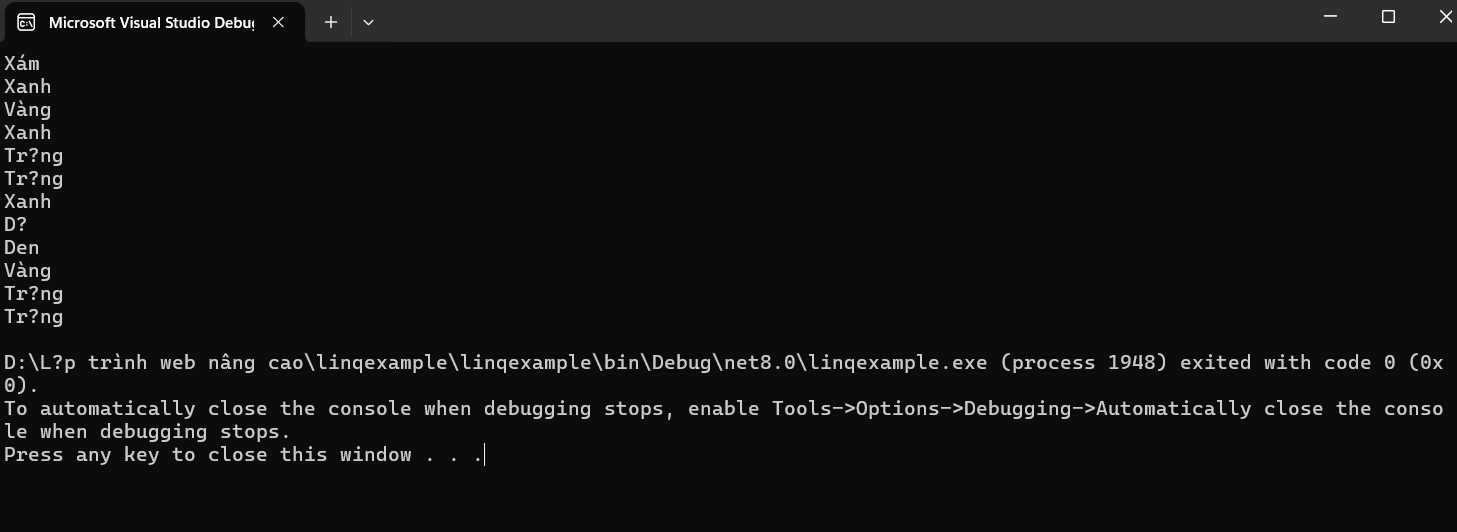




A screenshot of a computer

AI-generated content may be incorrect.





A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

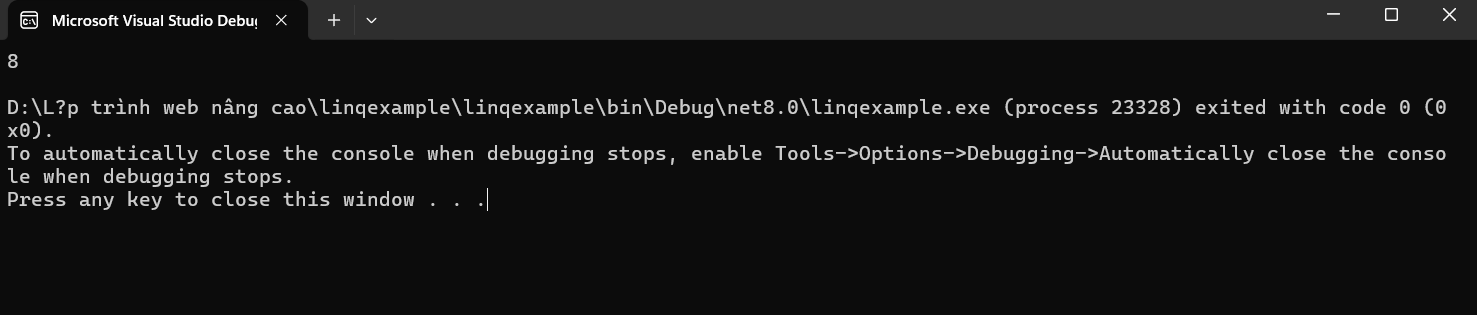
AI-generated content may be incorrect.

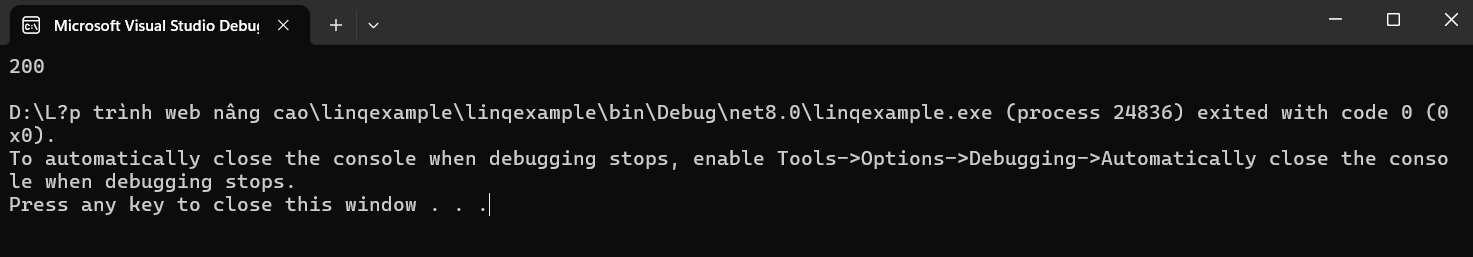
A screen shot of a computer

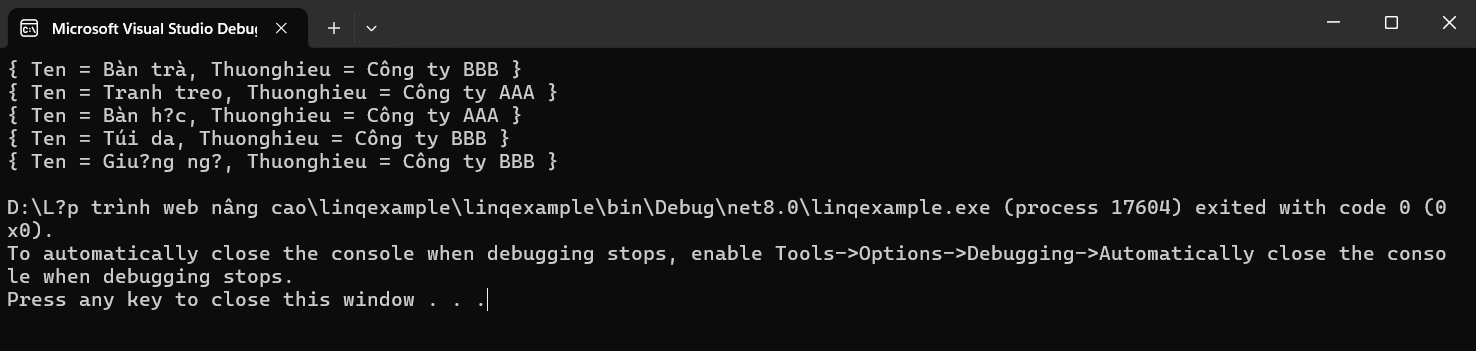
AI-generated content may be incorrect.

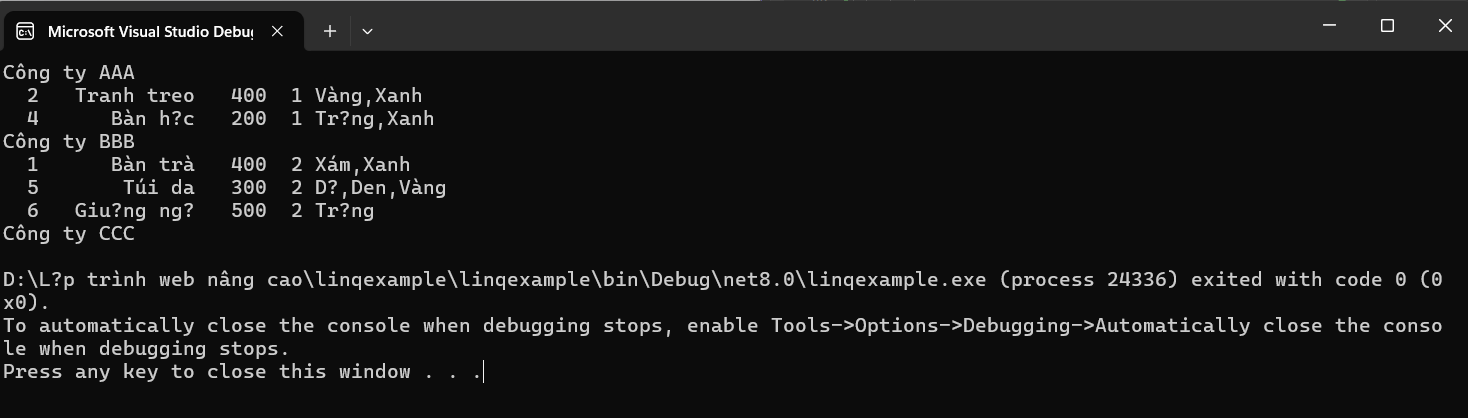
A screenshot of a computer

AI-generated content may be incorrect.



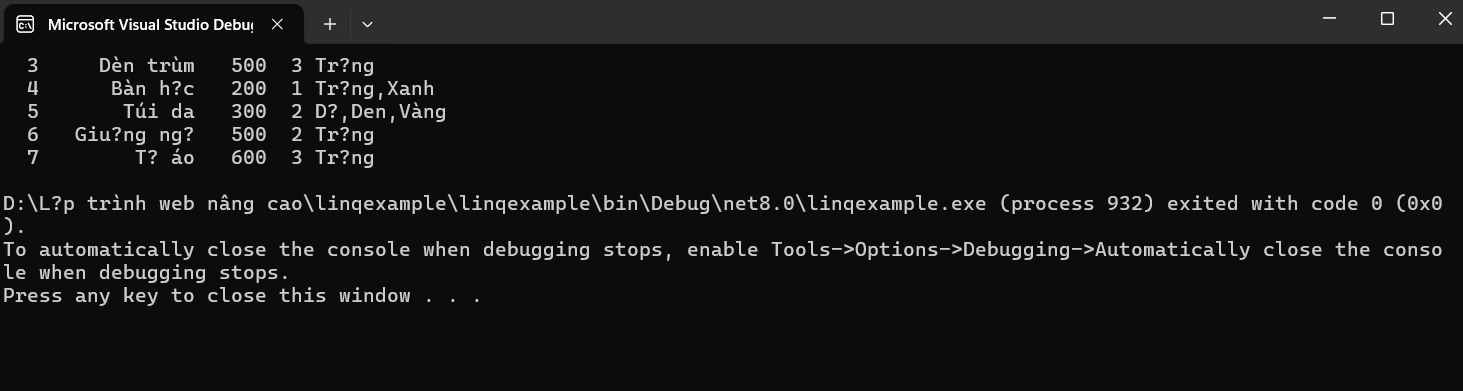


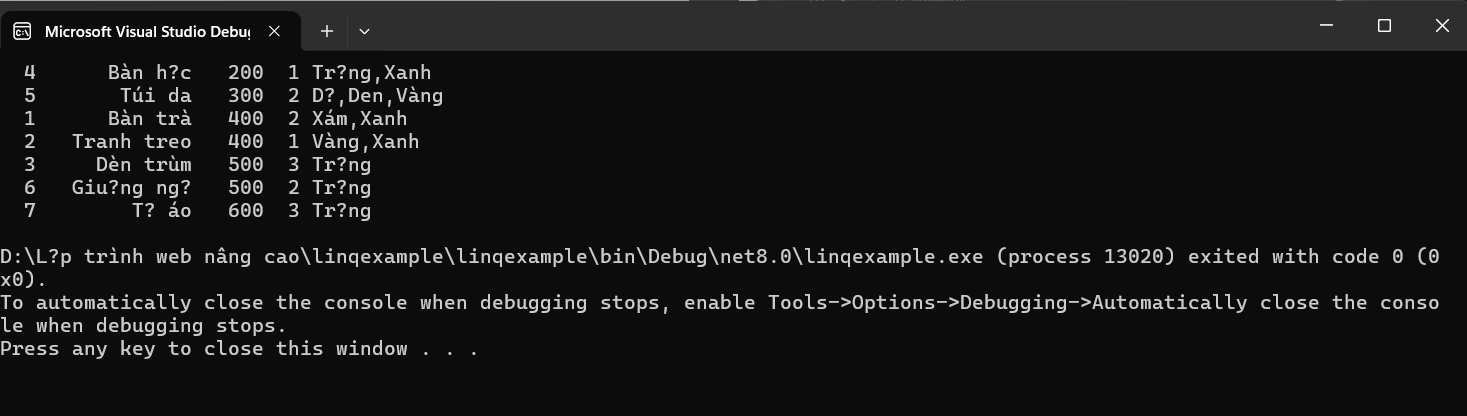


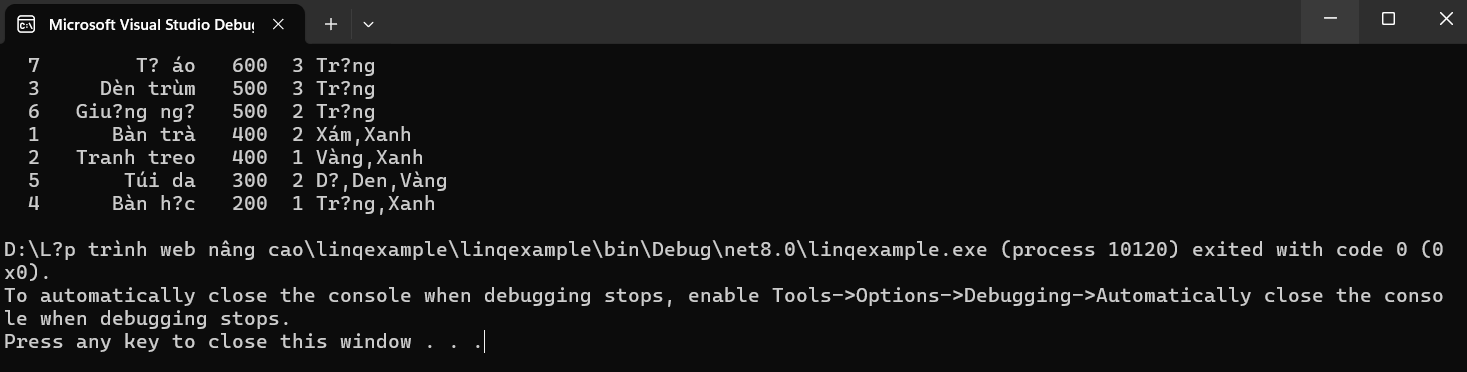


A screenshot of a computer

AI-generated content may be incorrect.







A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.