Họ và tên: Lê Hữu Hoàn

Lớp : Lập Trình C# (3)

Home work 29/9/2023

1 DELEGATE

A screen shot of a computer

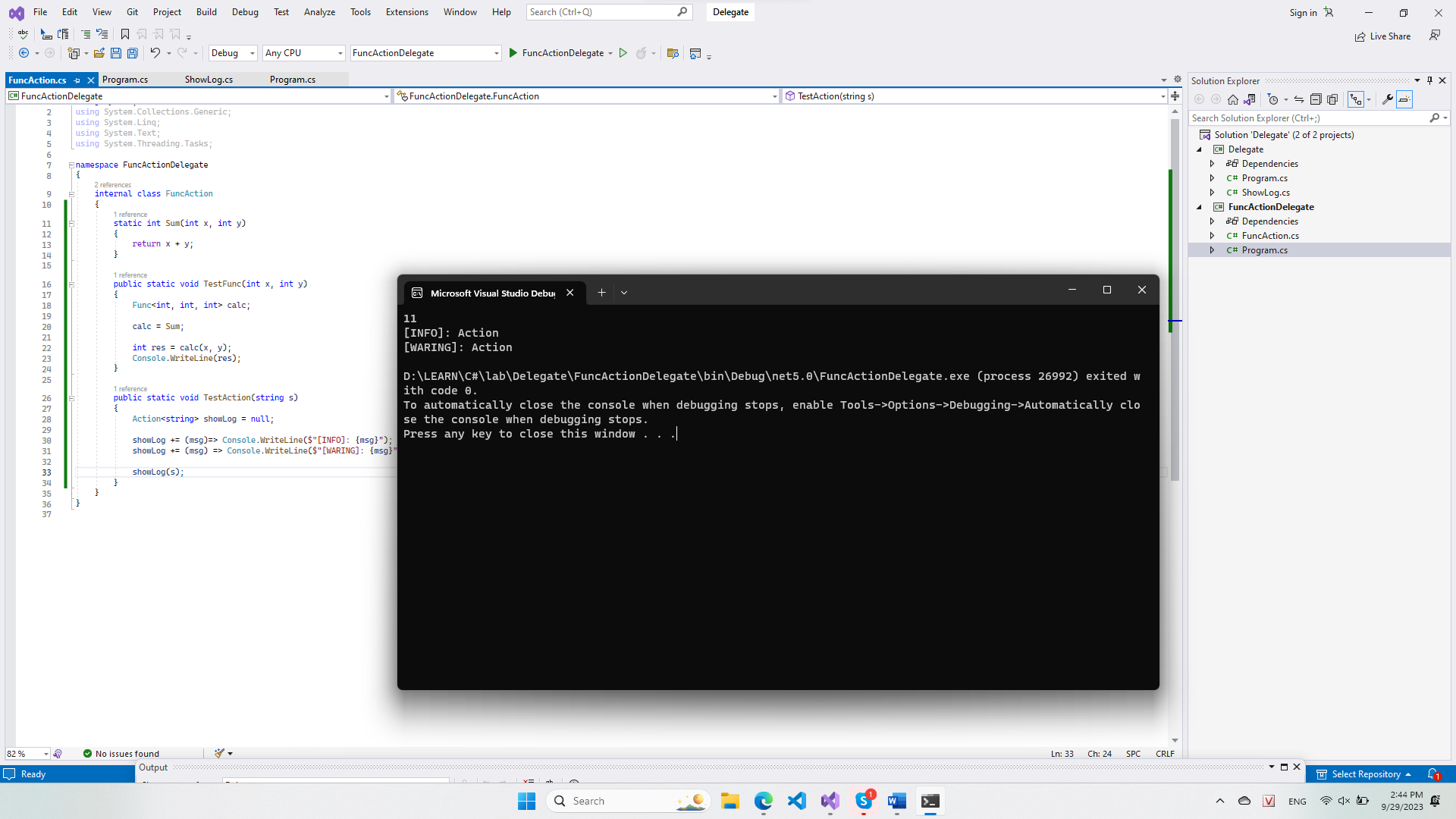
Description automatically generated

Code:

|  |
| --- |
| internal class Program  {  static void Main(string[] args)  {    ShowLogs.TestShowLog();  ShowLogs.TestShowLogMulti();  ShowLogs.TestShowLogPlus();  }  } |

|  |
| --- |
| internal class ShowLogs  {  public delegate void ShowLog(string message);  static public void Info(string msg)  {  Console.ForegroundColor = ConsoleColor.Green;  Console.WriteLine($"[INFO]: {msg}");  Console.ResetColor();  }  static public void Warning(string msg)  {  Console.ForegroundColor = ConsoleColor.Yellow;  Console.WriteLine($"[WARNING]: {msg}");  Console.ResetColor();  }  public static void TestShowLog()  {  ShowLog showLog;  showLog = Info;  showLog("Application started");  showLog = Warning;  showLog("System.Linq defined but not use");  }  public static void TestShowLogMulti()  {  ShowLog showLog = null;  showLog += Warning;  showLog += Info;  showLog += Warning;  showLog("Application log");  }  public static void TestShowLogPlus()  {  ShowLog showLog1 = (x) => { Console.WriteLine($"-----{x}-----"); };  ShowLog showLog2 = Warning;  ShowLog showLog3 = Info;  var all = showLog1 + showLog2 + showLog3 + showLog1;  all("Hi!");  }  } |

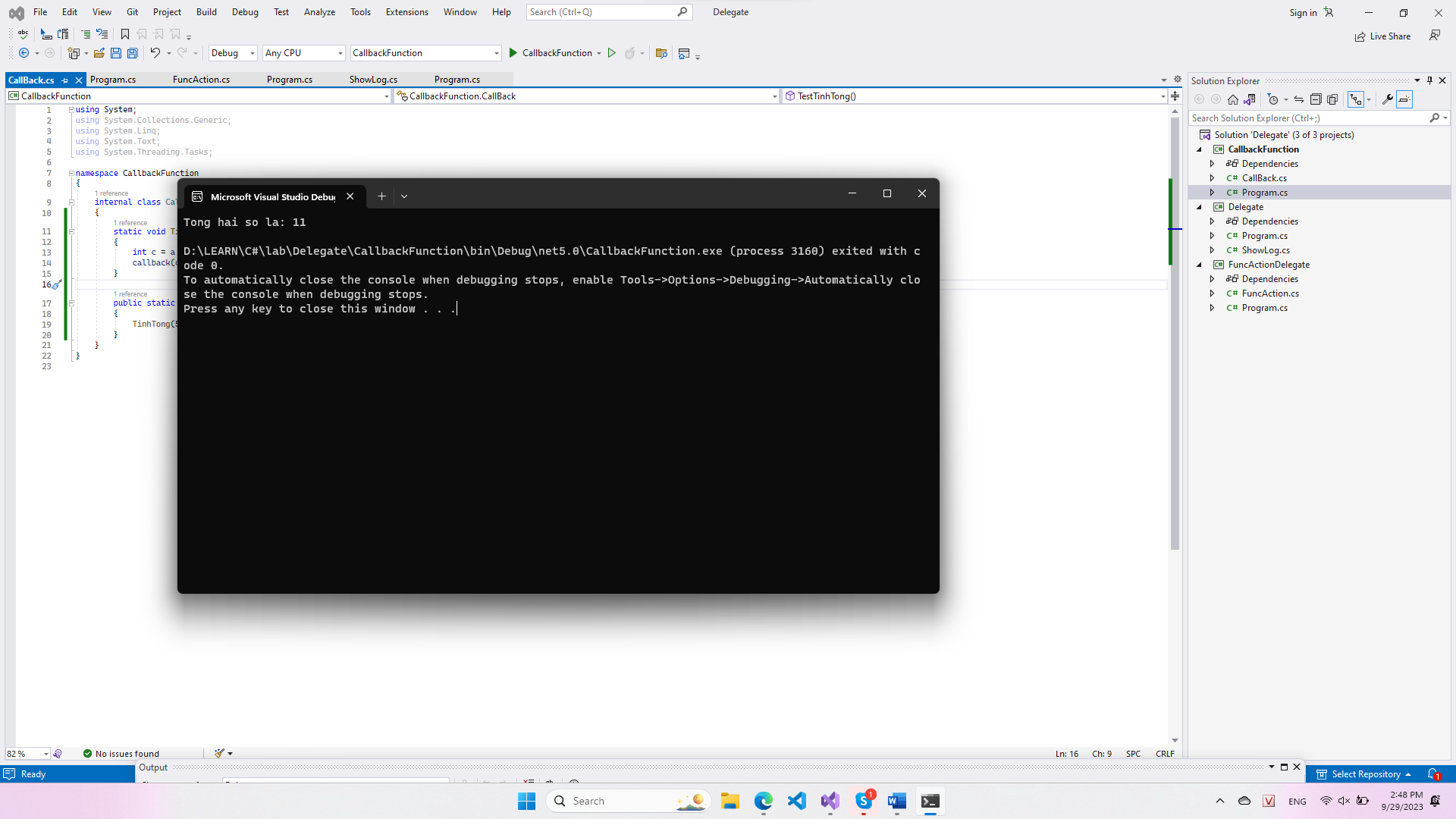
Func-Action



|  |
| --- |
| internal class FuncAction  {  static int Sum(int x, int y)  {  return x + y;  }  public static void TestFunc(int x, int y)  {  Func<int, int, int> calc;    calc = Sum;  int res = calc(x, y);  Console.WriteLine(res);  }  public static void TestAction(string s)  {  Action<string> showLog = null;  showLog += (msg)=> Console.WriteLine($"[INFO]: {msg}");  showLog += (msg) => Console.WriteLine($"[WARING]: {msg}");  showLog(s);  }  } |

|  |
| --- |
| internal class Program  {  static void Main(string[] args)  {  FuncAction.TestFunc(5, 6);  FuncAction.TestAction("Action");  }  } |

CALLBACK



|  |
| --- |
| internal class CallBack  {  static void TinhTong(int a, int b, Action<int> callback)  {  int c = a + b;  callback(c);  }  public static void TestTinhTong()  {  TinhTong(5, 6, (x) => Console.WriteLine($"Tong hai so la: {x}"));  }  } |

|  |
| --- |
| internal class Program  {  static void Main(string[] args)  {  CallBack.TestTinhTong();  }  } |

2 Lambda

A computer screen with a black screen

Description automatically generated

|  |
| --- |
| internal class Program  {  public delegate int TinhToan(int a, int b);  static void Main(string[] args)  {  // Gán biểu thức lambda cho delegate  TinhToan tinhtong = (int x, int y) => {  return x + y;  };  tinhtong += (int x, int y) => x + y;  int kq = tinhtong(5, 1);  Console.WriteLine(kq);  }  } |

3 Event

A computer screen shot of a black screen

Description automatically generated

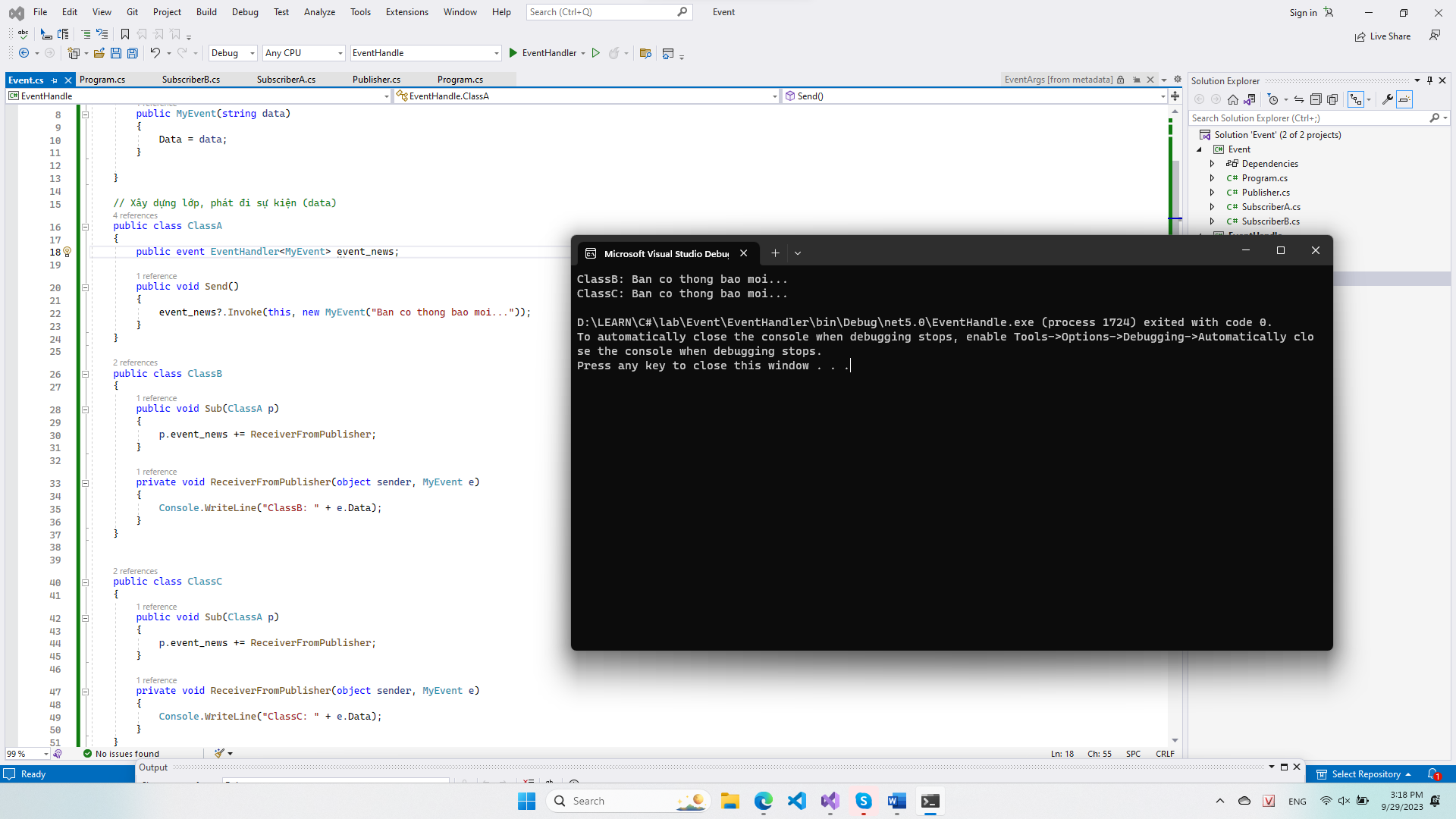
|  |
| --- |
| internal class SubscriberB  {  public void Sub(Publisher p)  {  p.event\_news += ReceiverFromPublisher;  }  void ReceiverFromPublisher(object data)  {  Console.WriteLine("SubscriberB: " + data.ToString());  }  } |

|  |
| --- |
| internal class SubscriberA  {  public void Sub(Publisher p)  {  p.event\_news += ReceiverFromPublisher;  }  void ReceiverFromPublisher(object data)  {  Console.WriteLine("SubscriberA: " + data.ToString());  }  } |

|  |
| --- |
| internal class Publisher  {  public delegate void NotifyNews(object data);  public NotifyNews event\_news;  public void Send()  {  event\_news?.Invoke("Ban co thong bao moi");  }  } |

|  |
| --- |
| internal class Program  {  static void Main(string[] args)  {  Publisher p = new Publisher();  SubscriberA sa = new SubscriberA();  SubscriberB sb = new SubscriberB();  sa.Sub(p);  sb.Sub(p);  p.Send();  }  } |

EVENT



|  |
| --- |
| public class MyEvent:EventArgs  {  public string Data{ get; }  public MyEvent(string data)  {  Data = data;  }  }  // Xây dựng lớp, phát đi sự kiện (data)  public class ClassA  {  public event EventHandler<MyEvent> event\_news;  public void Send()  {  event\_news?.Invoke(this, new MyEvent("Ban co thong bao moi..."));  }  }  public class ClassB  {  public void Sub(ClassA p)  {  p.event\_news += ReceiverFromPublisher;  }  private void ReceiverFromPublisher(object sender, MyEvent e)  {  Console.WriteLine("ClassB: " + e.Data);  }  }  public class ClassC  {  public void Sub(ClassA p)  {  p.event\_news += ReceiverFromPublisher;  }  private void ReceiverFromPublisher(object sender, MyEvent e)  {  Console.WriteLine("ClassC: " + e.Data);  }  } |

|  |
| --- |
| internal class Program  {  static void Main(string[] args)  {  ClassA p = new ClassA();  ClassB sa = new ClassB();  ClassC sb = new ClassC();  sa.Sub(p);  sb.Sub(p);  p.Send();  }  } |

Extension Methods

A computer screen shot of a black screen

Description automatically generated

|  |
| --- |
| internal class Program  {  public static void Print(string s, ConsoleColor color = ConsoleColor.Yellow)  {  ConsoleColor lastColor = Console.ForegroundColor;  Console.ForegroundColor = color;  Console.WriteLine(s);  Console.ForegroundColor = lastColor;  }  static void Main(string[] args)  {  string s = "test string";  s.Print();  "Hello world".Print(ConsoleColor.Red);  }  } |

|  |
| --- |
| internal static class MyExtensionMethods  {  public static void Print(this string s, ConsoleColor color = ConsoleColor.Yellow)  {  ConsoleColor lastColor = Console.ForegroundColor;  Console.ForegroundColor = color;  Console.WriteLine(s);  Console.ForegroundColor = lastColor;  }  } |

LinQ

A screenshot of a computer

Description automatically generated

|  |
| --- |
| internal class Program  {  static void Main(string[] args)  {  Products.ProductPrice500();    var products = Products.products;  var brands = Brand.brands;  // truy vấn cơ bản  var ketqua = from product in products  where product.Price == 400  select product;  Console.WriteLine("cac san pham co gia 400:");  foreach (var product in ketqua)  Console.WriteLine(product.ToString());  Console.WriteLine();  // tạo đối tượng vô danh kết quả trả về  var ketqua1 = from product in products  where product.Price == 400  select new {  ten = product.Name,  mausac = string.Join(',', product.Colors)  };  Console.WriteLine("Ten, mau san pham co gia 400");  foreach (var item in ketqua1) Console.WriteLine(item.ten + " - " + item.mausac);  Console.WriteLine();  // lọc dữ liệu bằng where  var ketqua2 = from product in products  where product.Price >= 500  where product.Name.StartsWith("Giuong")  select product;  Console.WriteLine("San pham co ten bat dau la Giuong, gia tren 500");  foreach (var product in ketqua2)  Console.WriteLine(product.ToString());  Console.WriteLine();  //sử dụng from kết hợp để lọc  var ketqua3 = from product in products  from color in product.Colors  where product.Price < 500  where color == "Vang"  select product;  Console.WriteLine("San pham co mau vang , gia duoi 500");  foreach (var product in ketqua3) Console.WriteLine(product.ToString());  Console.WriteLine();  // Sắp xếp bằng orderby  var ketqua4 = from product in products  where product.Price <= 300  orderby product.Price descending  select product;  Console.WriteLine("San pham có gia duoi 300 sap xep theo gia nho dan");  foreach (var product in ketqua4) Console.WriteLine($"{product.Name} - {product.Price}");  Console.WriteLine();  // Nhóm kết quả bằng group  var ketqua5 = from product in products  where product.Price >=400 && product.Price <= 500  group product by product.Price;  Console.WriteLine("Cac san pham theo nhom gia 400, 500");  foreach (var group in ketqua5)  {  Console.WriteLine(group.Key);  foreach (var product in group)  {  Console.WriteLine($" {product.Name} - {product.Price}");  }  }  Console.WriteLine();  // dùng biến trong truy vấn  var ketqua6 = from product in products  group product by product.Price into gr  let count = gr.Count()  select new {  price = gr.Key,  number\_product = count  };  Console.WriteLine("Cac san pham theo gia");  foreach (var item in ketqua6)  {  Console.WriteLine($" Giá {item.price} - có {item.number\_product} sp");  }  Console.WriteLine();  // inner join  var ketqua7 = from product in products  join brand in brands on product.Brand equals brand.ID  select new {  name = product.Name,  brand = brand.Name,  price = product.Price  };  Console.WriteLine("San pham- gia- ten hang");  foreach (var item in ketqua7)  {  Console.WriteLine($"{item.name,10} {item.price, 4} {item.brand,12}");  }  Console.WriteLine();  // left join  var ketqua8 = from product in products  join brand in brands on product.Brand equals brand.ID into t  from brand in t.DefaultIfEmpty()  select new {  name = product.Name,  brand = (brand == null) ? "NO-BRAND" : brand.Name,  price = product.Price  };  Console.WriteLine("San pham- gia -ten hang");  foreach (var item in ketqua8)  {  Console.WriteLine($"{item.name,10} {item.price, 4} {item.brand,12}");  }      }  } |

|  |
| --- |
| internal class Products  {  public static List<Product> products;  static Products()  {  products = new List<Product>()  {  new Product(1, "Ban",200, new string[] {"Trang", "Xanh"},1),  new Product(2, "tui",300, new string[] {"Do", "Den", "Vang"},2),  new Product(3, "Ban tra",400, new string[] {"Xam", "Xanh"},2),  new Product(4, "Tranh treo", 400, new string[] {"Vang", "Xanh"},1),  new Product(5, "Den",500, new string[] {"Trang"},3),  new Product(6, "Giuong", 500, new string[] {"Trang"},2),  new Product(7, "Tu",600, new string[] {"Trang"},3),  };  }  public static void ProductPrice500()  {  var products = Products.products;  var ketqua = from product in products  where product.Price == 500  select product;  foreach (var product in ketqua)  Console.WriteLine(product.ToString());  }  } |

|  |
| --- |
| internal class Product  {  public int ID { set; get; }  public string Name { set; get; }  public double Price { set; get; }  public string[] Colors { set; get; }  public int Brand { set; get; }  public Product(int id, string name, double price, string[] colors, int brand)  {  ID = id; Name = name; Price = price; Colors = colors; Brand = brand;  }  override public string ToString() => $"{ID,3} {Name,12} {Price,5} {Brand,2} {string.Join(",", Colors)}";  } |

|  |
| --- |
| public class Brand  {  public string Name { set; get; }  public int ID { set; get; }  static List<Brand> \_brands;  public static List<Brand> brands  {  get  {  if (\_brands == null)  {  \_brands = new List<Brand>() {  new Brand{ID = 1, Name = "Company 1"},  new Brand{ID = 2, Name = "Company 2"},  new Brand{ID = 4, Name = "Company 3"},  };  }  return \_brands;  }  }  } |