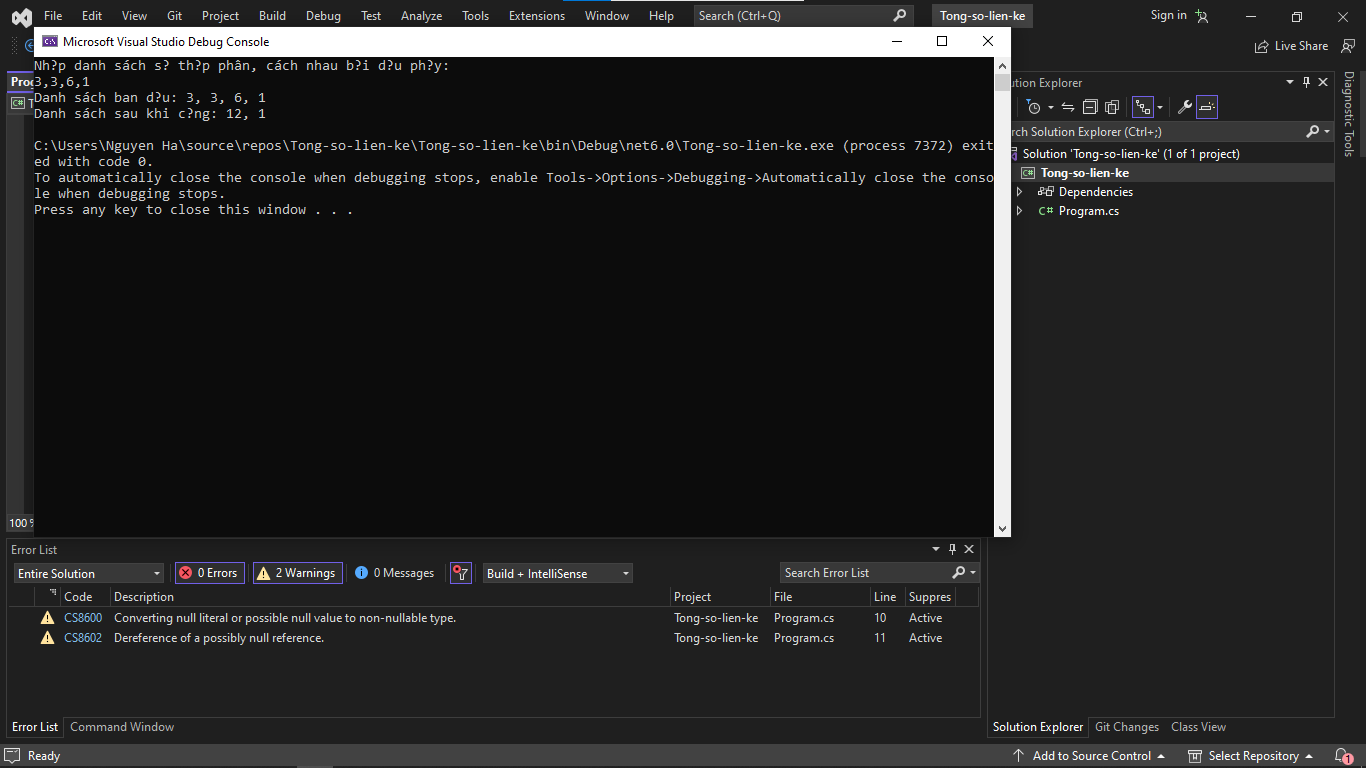
Họ và tên: Hồ Nguyên Hà . Mã SV: 20it043

Bài Tập Tuần 5

**Ex1:**



using System;

using System.Collections.Generic;

using System.Linq;

class Program

{

static void Main()

{

Console.WriteLine("Nhập danh sách số thập phân, cách nhau bởi dấu phẩy:");

string input = Console.ReadLine();

List<decimal> numbers = input.Split(',').Select(decimal.Parse).ToList();

Console.WriteLine("Danh sách ban đầu: " + string.Join(", ", numbers));

for (int i = 0; i < numbers.Count - 1; i++)

{

if (numbers[i] == numbers[i + 1])

{

numbers[i] += numbers[i + 1];

numbers.RemoveAt(i + 1);

i--;

}

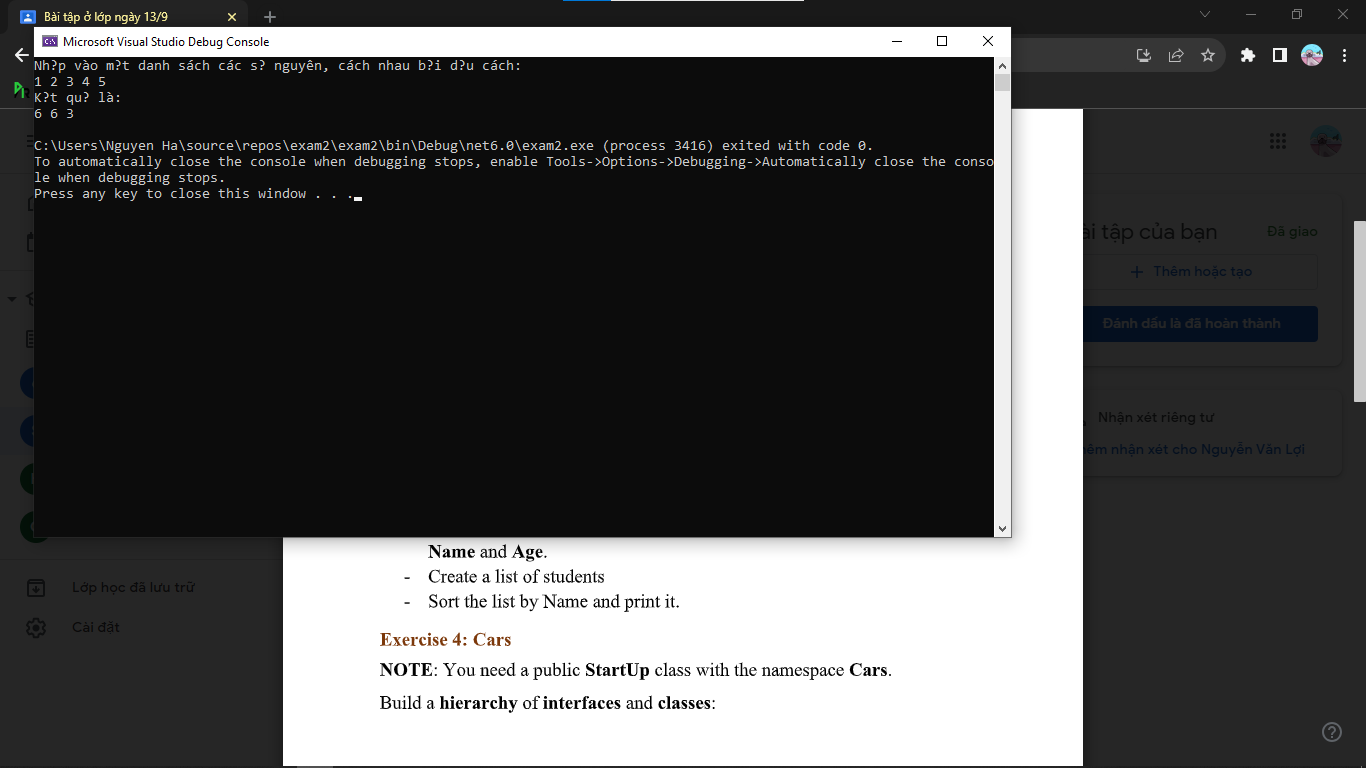
}

Console.WriteLine("Danh sách sau khi cộng: " + string.Join(", ", numbers));

}

}

**Ex2:**



using System;

using System.Collections.Generic;

using System.Linq;

class Program

{

static void Main(string[] args)

{

// Nhập vào một danh sách các số nguyên, cách nhau bởi dấu cách

Console.WriteLine("Nhập vào một danh sách các số nguyên, cách nhau bởi dấu cách:");

string input = Console.ReadLine();

// Chuyển đổi chuỗi thành mảng các số nguyên

int[] numbers = input.Split().Select(int.Parse).ToArray();

// Tạo một danh sách để lưu kết quả

List<int> result = new List<int>();

// Lặp cho đến khi số đầu không vượt quá số cuối

for (int i = 0; i < numbers.Length / 2; i++)

{

int currentSum = numbers[i] + numbers[numbers.Length - 1 - i];

result.Add(currentSum);

}

// Nếu danh sách có độ dài lẻ, thêm số ở giữa vào kết quả

if (numbers.Length % 2 == 1)

{

result.Add(numbers[numbers.Length / 2]);

}

// In ra kết quả, cách nhau bởi dấu cách

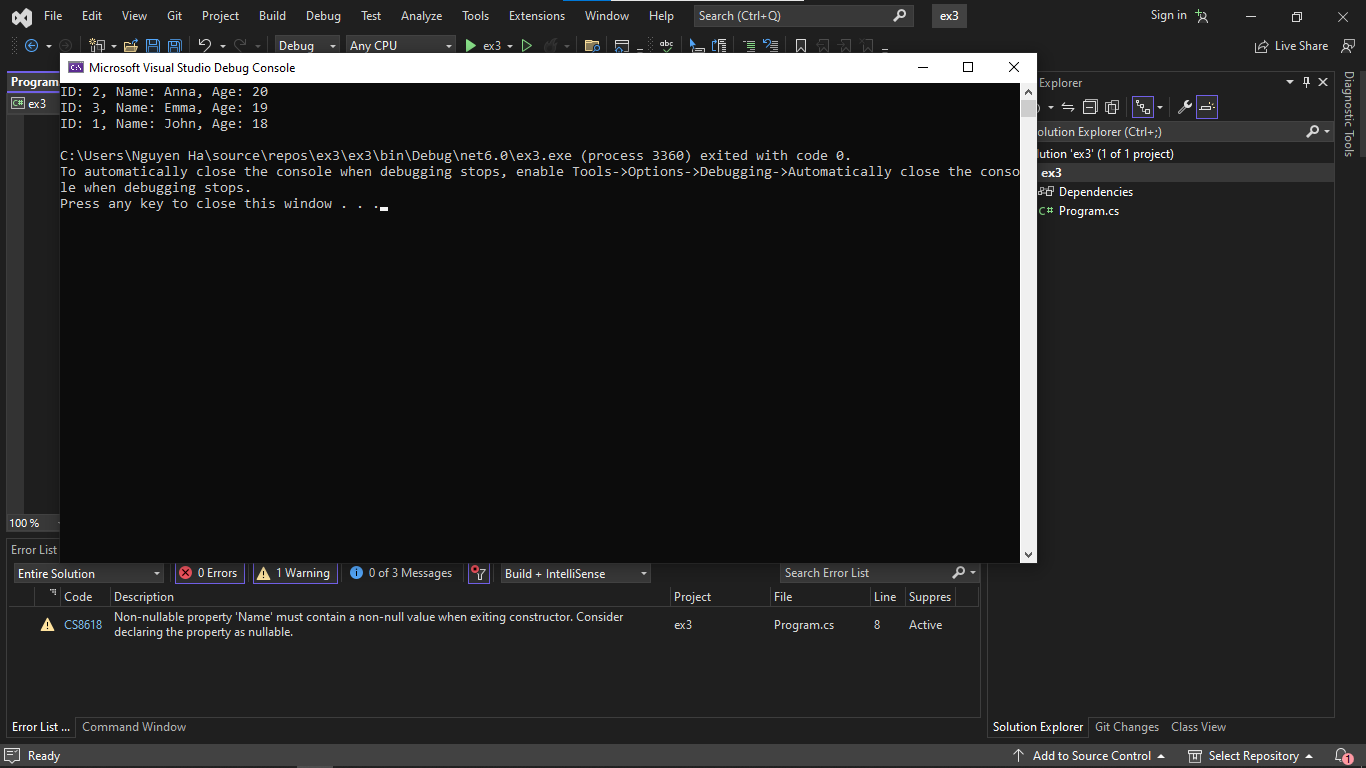
Console.WriteLine("Kết quả là:");

Console.WriteLine(string.Join(" ", result));

}

}

**Ex3:**



using System;

using System.Collections.Generic;

using System.Linq;

public class Student

{

public int ID { get; set; }

public string Name { get; set; }

public int Age { get; set; }

}

class Program

{

static void Main(string[] args)

{

// Create a list of students

List<Student> students = new List<Student>()

{

new Student() { ID = 1, Name = "John", Age = 18 },

new Student() { ID = 2, Name = "Anna", Age = 20 },

new Student() { ID = 3, Name = "Emma", Age = 19 }

};

// Sort the list by Name

var sortedStudents = students.OrderBy(s => s.Name).ToList();

// Print the sorted list

foreach (var student in sortedStudents)

{

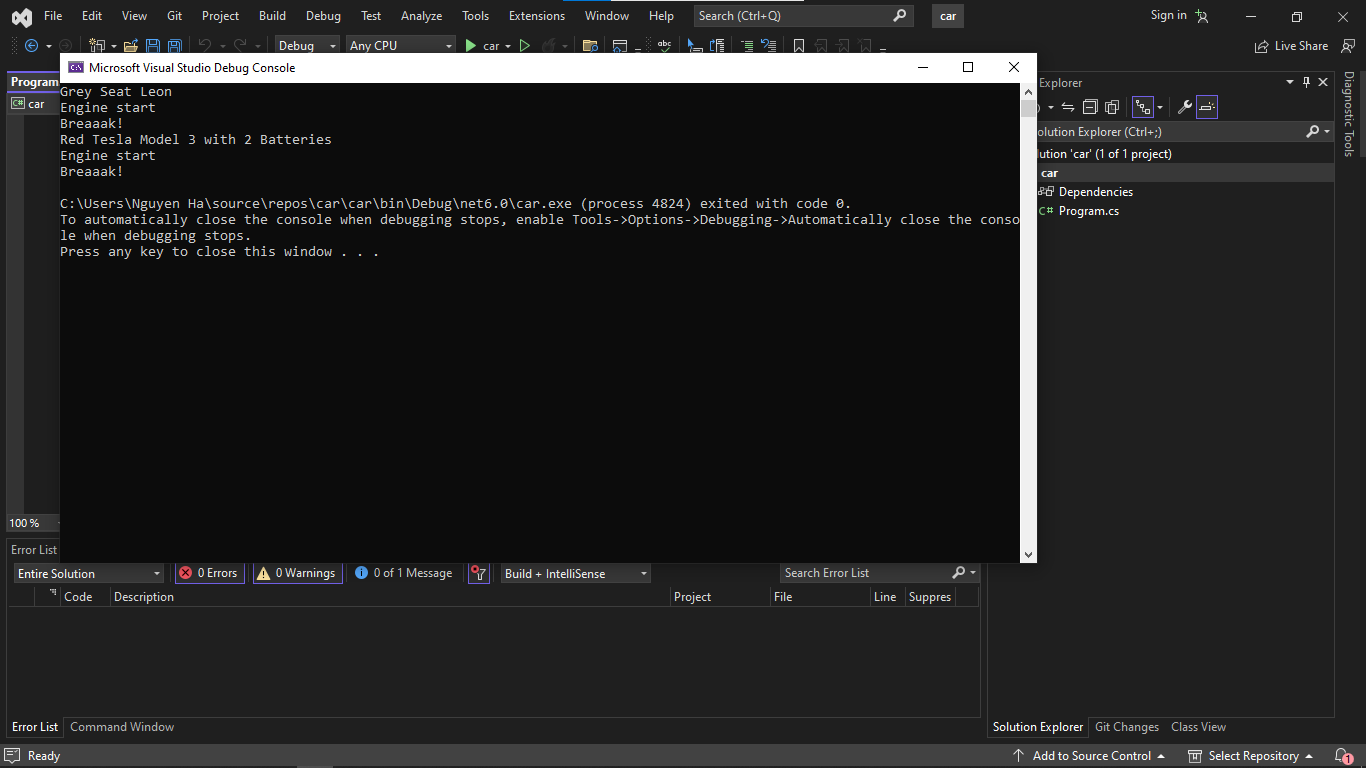
Console.WriteLine($"ID: {student.ID}, Name: {student.Name}, Age: {student.Age}");

}

}

}

**Ex4:**



namespace Cars

{

public interface ICar

{

string Model { get; set; }

string Color { get; set; }

void Start();

void Stop();

}

public class Car : ICar

{

public string Model { get; set; }

public string Color { get; set; }

public Car(string model, string color)

{

Model = model;

Color = color;

}

public virtual void Start()

{

Console.WriteLine("Engine start");

}

public virtual void Stop()

{

Console.WriteLine("Breaaak!");

}

public override string ToString()

{

return $"{Color} Car {Model}";

}

}

public interface IElectricCar : ICar

{

int Battery { get; set; }

}

public class ElectricCar : Car, IElectricCar

{

public int Battery { get; set; }

public ElectricCar(string model, string color, int battery) : base(model, color)

{

Battery = battery;

}

}

public class Tesla : ElectricCar

{

public Tesla(string model, string color, int battery) : base(model, color, battery) { }

public override string ToString()

{

return $"{Color} Tesla {Model} with {Battery} Batteries";

}

}

public class Seat : Car

{

public Seat(string model, string color) : base(model, color) { }

public override string ToString()

{

return $"{Color} Seat {Model}";

}

}

public class StartUp

{

static void Main(string[] args)

{

ICar seat = new Seat("Leon", "Grey");

Console.WriteLine(seat.ToString());

seat.Start();

seat.Stop();

ICar tesla = new Tesla("Model 3", "Red", 2);

Console.WriteLine(tesla.ToString());

tesla.Start();

tesla.Stop();

}

}

}