HOMEWORK: Collection

NAME: Hoàng Trung Hiếu

CLASS: 20Se3

Ex1:

Code :

using System;

using System.Collections.Generic;

using System.Linq;

class Program

{

static void Main(string[] args)

{

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

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

string input = Console.ReadLine();

// Chuyển đổi chuỗi thành một mảng các số thập phân

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

// Tạo một danh sách để lưu kết quả

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

// Duyệt qua mảng các số thập phân từ trái sang phải

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

{

// Lấy số hiện tại

decimal current = numbers[i];

// Nếu số hiện tại bằng số kế tiếp, thì cộng chúng lại

while (i < numbers.Length - 1 && current == numbers[i + 1])

{

current += numbers[i + 1];

i++;

}

// Thêm số hiện tại vào danh sách kết quả

result.Add(current);

}

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

Console.WriteLine("Kết quả là:");

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

}

}

KQ:

A computer screen with white text

Description automatically generated

Ex2:

Code :

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));

}

}

KQ:

A computer screen with a black screen

Description automatically generated

Ex3:

Code :

using System;

using System.Collections.Generic;

using System.Linq;

// Định nghĩa lớp Student

class Student

{

// Khai báo các thuộc tính ID, Name và Age kiểu int, string và int

public int ID { get; set; }

public string Name { get; set; }

public int Age { get; set; }

// Khởi tạo lớp Student với ba tham số id, name và age

public Student(int id, string name, int age)

{

ID = id;

Name = name;

Age = age;

}

// Ghi đè phương thức ToString để trả về chuỗi biểu diễn thông tin của sinh viên

public override string ToString()

{

return $"ID: {ID}, Name: {Name}, Age: {Age}";

}

}

class Program

{

static void Main(string[] args)

{

// Tạo một danh sách các sinh viên

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

// Thêm vào danh sách một số sinh viên mẫu

students.Add(new Student(1, "An", 20));

students.Add(new Student(2, "Binh", 19));

students.Add(new Student(3, "Chau", 21));

students.Add(new Student(4, "Dung", 18));

students.Add(new Student(5, "Eva", 22));

// Sắp xếp danh sách theo Name bằng phương thức OrderBy của LINQ

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

// In ra màn hình danh sách các sinh viên sau khi sắp xếp

Console.WriteLine("Danh sách các sinh viên sau khi sắp xếp theo Name:");

foreach (Student s in students)

{

Console.WriteLine(s);

}

}

}

KQ:

A computer screen with white text

Description automatically generated

Ex4:

Code :

using System;

// Khai báo không gian tên Car

namespace ElectricCarExample

{

// Khai báo giao diện ICar

public interface ICar

{

// Khai báo thuộc tính Model kiểu string

string Model { get; set; }

// Khai báo thuộc tính Color kiểu string

string Color { get; set; }

// Khai báo phương thức Start kiểu string

string Start();

// Khai báo phương thức Stop kiểu string

string Stop();

// Khai báo phương thức ToString kiểu string

string ToString();

}

// Khai báo lớp ElectricCar thực thi giao diện ICar

public class ElectricCar : ICar

{

// Khai báo và triển khai thuộc tính Model kiểu string

public string Model { get; set; }

// Khai báo và triển khai thuộc tính Color kiểu string

public string Color { get; set; }

// Khởi tạo lớp ElectricCar với hai tham số model và color

public ElectricCar(string model, string color)

{

Model = model;

Color = color;

}

// Triển khai phương thức Start kiểu string, trả về chuỗi "Engine start"

public virtual string Start()

{

return "Engine start";

}

// Triển khai phương thức Stop kiểu string, trả về chuỗi "Breaaak!"

public string Stop()

{

return "Breaaak!";

}

// Triển khai phương thức ToString kiểu string

public override string ToString()

{

return $"{Color} {GetType().Name} {Model}";

}

}

// Khai báo lớp Seat kế thừa từ lớp ElectricCar

public class Seat : ElectricCar

{

// Khai báo và triển khai thuộc tính Seats kiểu int

public int Seats { get; set; }

// Khởi tạo lớp Seat với ba tham số model, color, và seats, gọi đến hàm khởi tạo của lớp cha ElectricCar

public Seat(string model, string color, int seats) : base(model, color)

{

Seats = seats;

}

// Ghi đè phương thức ToString kiểu string

public override string ToString()

{

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

}

}

// Khai báo lớp Tesla kế thừa từ lớp ElectricCar

public class Tesla : ElectricCar

{

// Khai báo và triển khai thuộc tính Batteries kiểu int

public int Batteries { get; set; }

// Khởi tạo lớp Tesla với ba tham số model, color, và batteries, gọi đến hàm khởi tạo của lớp cha ElectricCar

public Tesla(string model, string color, int batteries) : base(model, color)

{

Batteries = batteries;

}

// Ghi đè phương thức ToString kiểu string

public override string ToString()

{

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

}

}

class Program

{

static void Main(string[] args)

{

ICar seat = new Seat("Leon", "Grey", 5);

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

Console.WriteLine(seat.ToString());

Console.WriteLine(seat.Start());

Console.WriteLine(seat.Stop());

Console.WriteLine(tesla.ToString());

Console.WriteLine(tesla.Start());

Console.WriteLine(tesla.Stop());

}

}

}

KQ:

A computer screen with white text

Description automatically generated