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

Bài Tập Tuần 3

**Ex1:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Lab01

{

class Student

{

private int SID;

private string TenSV;

private string Khoa;

private float DiemTB;

public Student() //Constructor mac dinh

{

SID = 1;

TenSV = "Nguyen Van Nam";

Khoa = "CNTT";

DiemTB = 7;

}

public Student(Student stu) //Constructor sao chep

{

SID = stu.SID;

TenSV = stu.TenSV;

Khoa = stu.Khoa;

DiemTB =stu.DiemTB;

}

//Constructor tham so

public Student(int id, string ten, string kh, float dtb)

{

SID = id;

TenSV = ten;

Khoa = kh;

DiemTB = dtb;

}

//Cac Property cho tung thuoc tinh cua lop

public int StudentID //Property dai dien cho thuoc tinh SID

{

get { return SID; } //Lay du lieu

set { SID = value;} //Gan du lieu

}

public String Name

{

get{ return TenSV; }

set{ TenSV = value; }

}

public String Faculty

{

get { return Khoa; }

set { Khoa = value; }

}

public float Mark

{

get { return DiemTB; }

set { DiemTB = value; }

}

//Phuong thuc hien thi du lieu

public void Show()

{

Console.WriteLine("MSSV:{0}", this.SID);

Console.WriteLine("Ten SV:{0}", this.TenSV);

Console.WriteLine("Khoa:{0}", this.Khoa);

Console.WriteLine("Diem TB:{0}", this.DiemTB);

}

}

class Tester

{

public static void Main()

{

Student[] DSSV;

int n;

Console.Write("Nhap so luong SV:");

n = int.Parse(Console.ReadLine());

DSSV = new Student[n]; //tao mang n phan tu

Console.WriteLine("\n ====NHAP DS SINH VIEN====");

for (int i = 0; i < n; i++) //Lap n lan nhap thong tin sv

{

DSSV[i] = new Student();

Console.Write("Nhap MaSV {0}:",i+1);

DSSV[i].StudentID =int.Parse(Console.ReadLine());

Console.Write("Ho ten SV:");

DSSV[i].Name = Console.ReadLine();

Console.Write("Nhap khoa:");

DSSV[i].Faculty = Console.ReadLine();

Console.Write("Nhap Diem TB:");

DSSV[i].Mark = float.Parse(Console.ReadLine());

}

//Xuat DS Sinh vien

Console.WriteLine("\n ====XUAT DS SINH VIEN====");

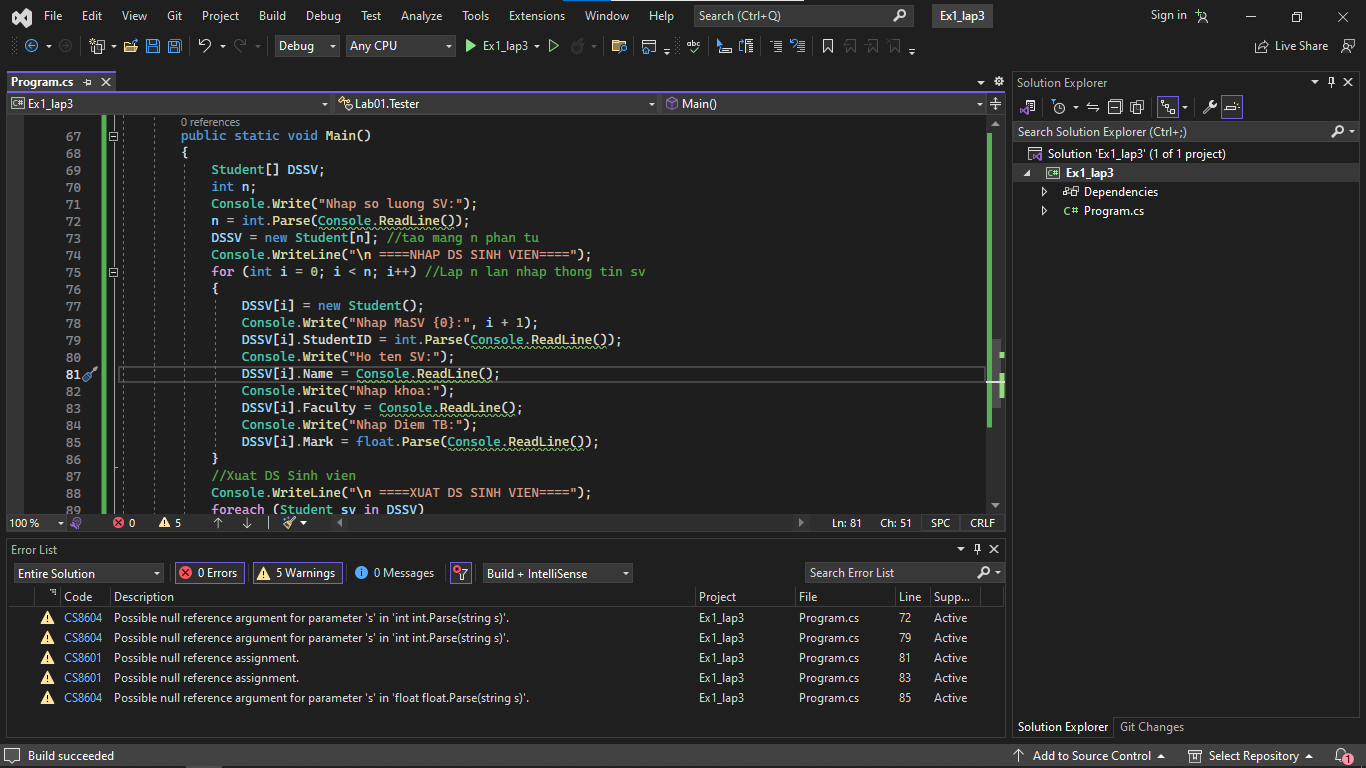
foreach (Student sv in DSSV)

sv.Show();

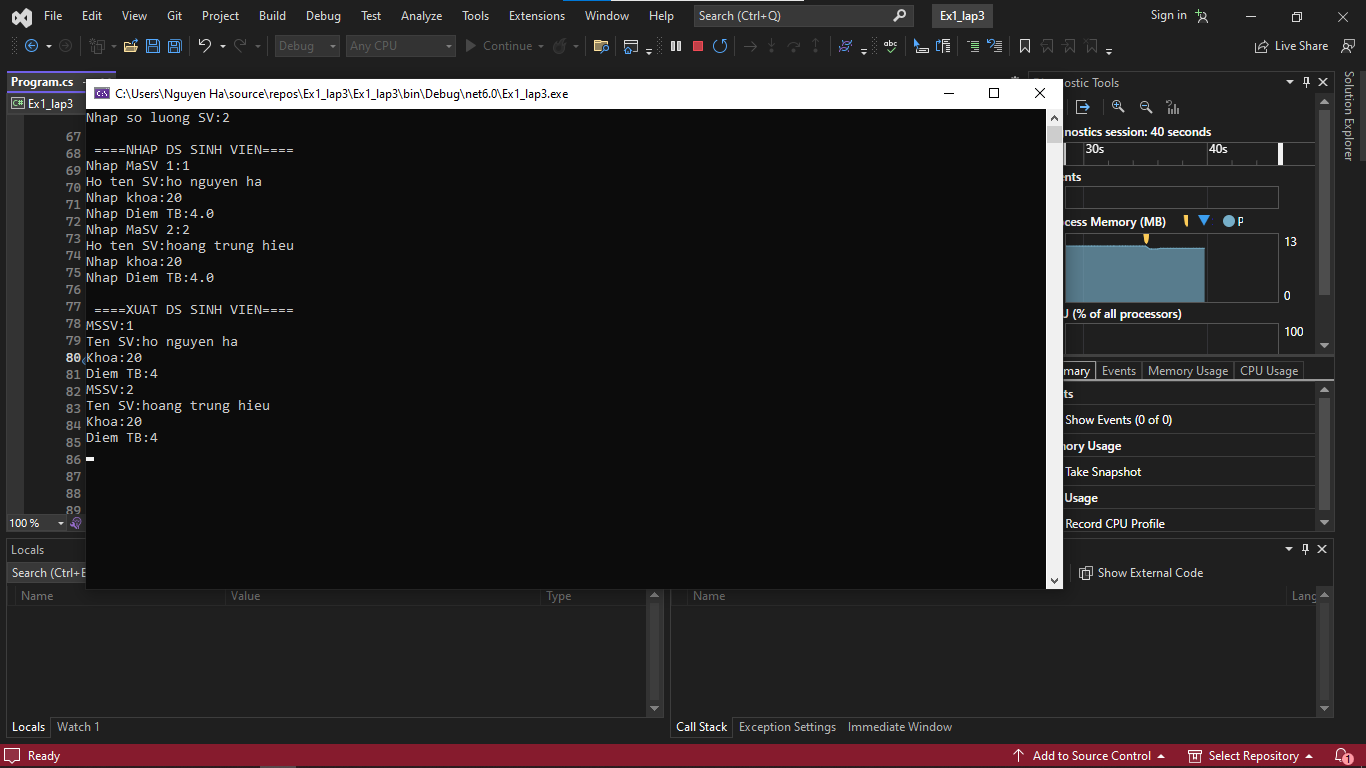
Console.ReadLine();

}

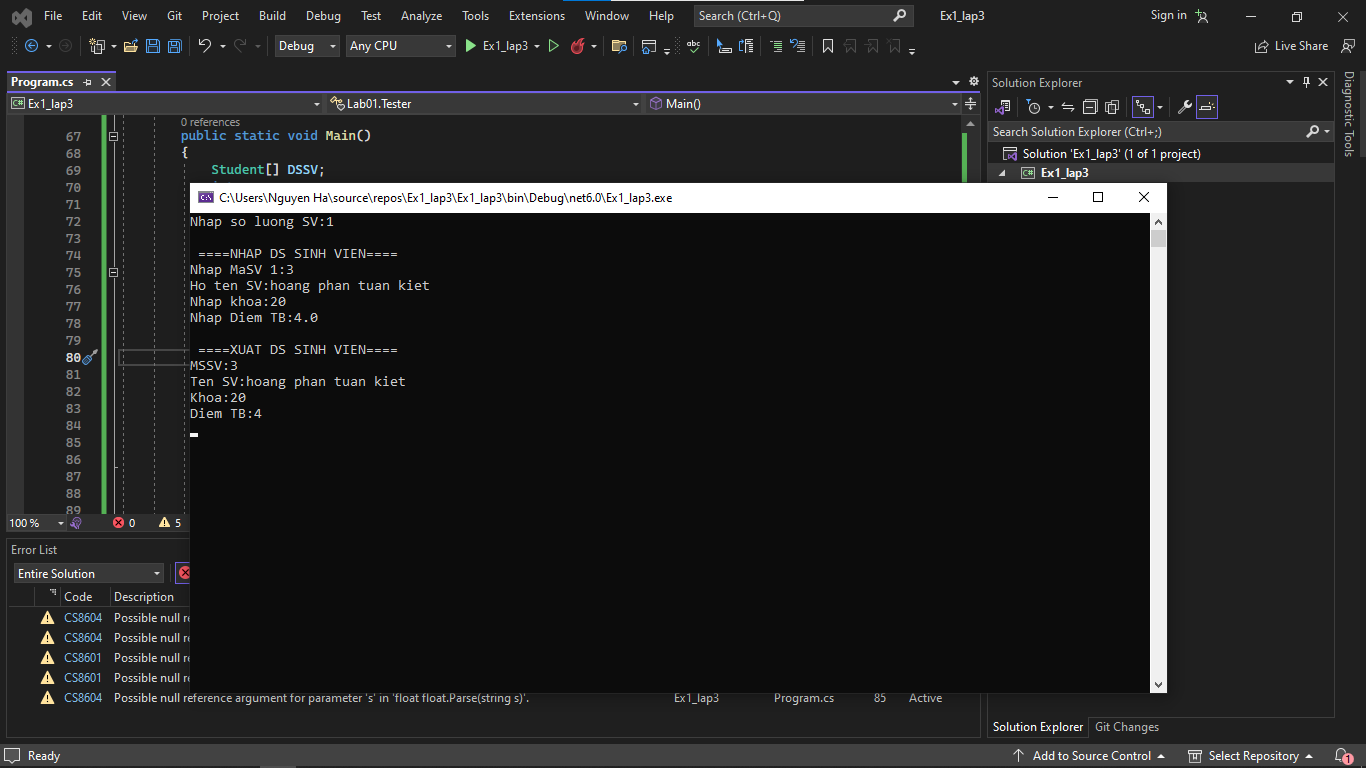
-Chức năng Build Solution: F6



-Chức năng Run with Debug: F5

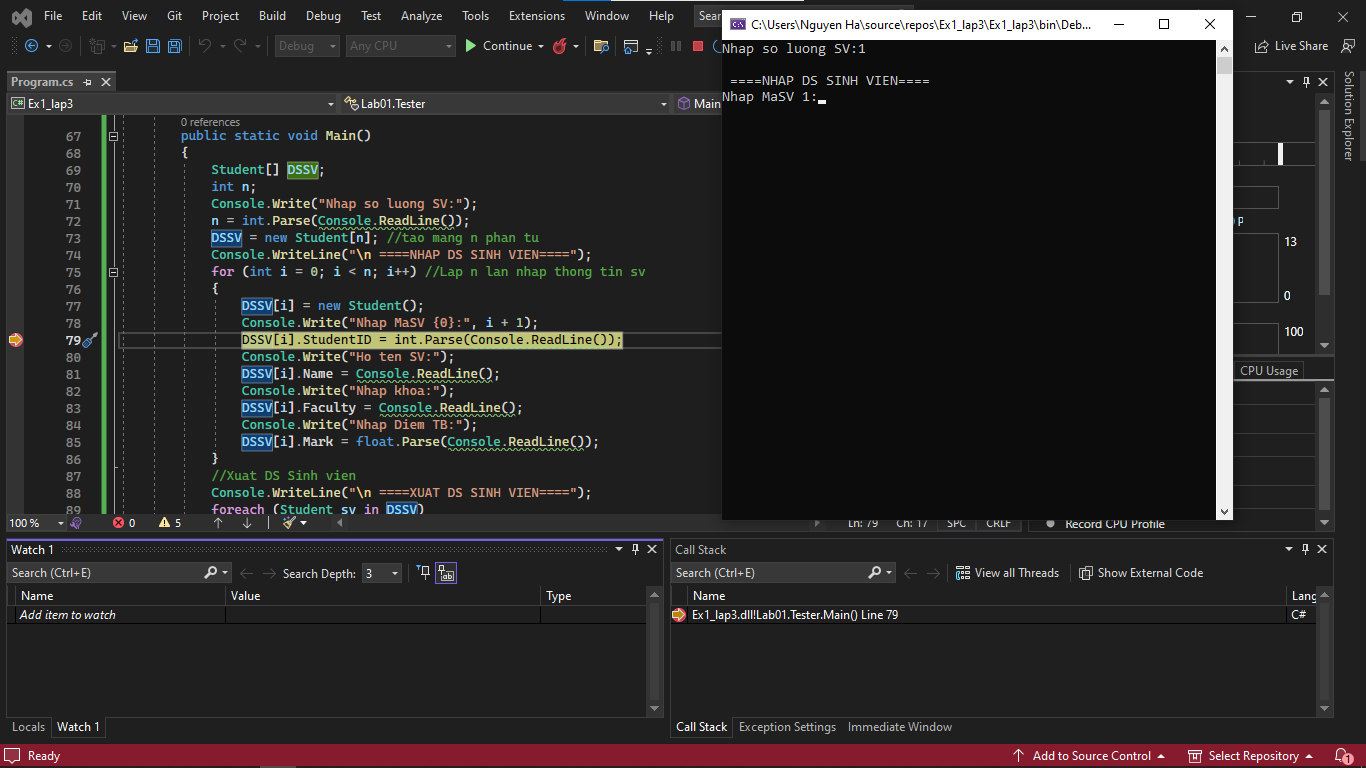


-Chức năng Run without Debug: Ctrl + F5

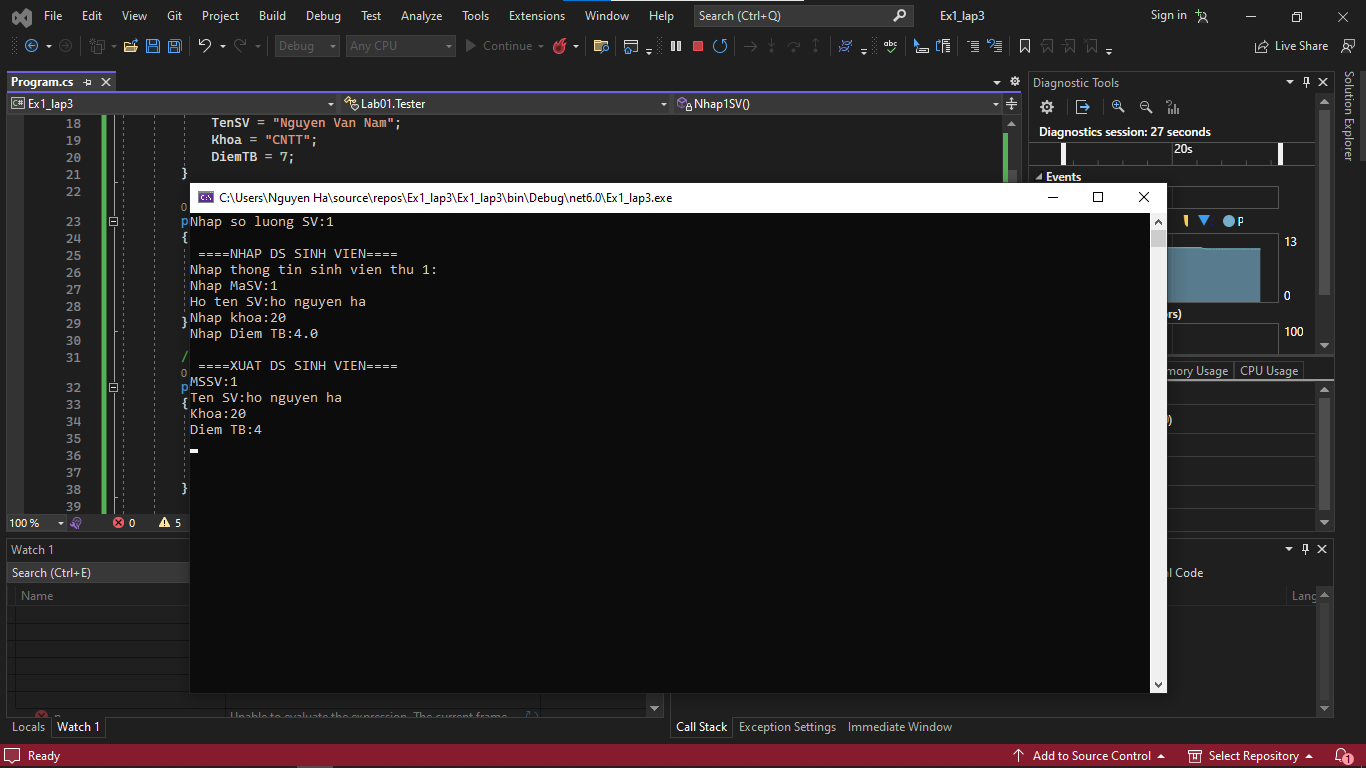


-Chạy từng bước chương trình trong chế độ debug: dùng breakpoint hoặc chạy từng dòng

lệnh. Kiểm tra những giá trị của các biến trong chương trình ở cửa sổ Watch.



**Ex2:**



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Lab01

{

class Student

{

private int SID;

private string TenSV;

private string Khoa;

private float DiemTB;

public Student() //Constructor mac dinh

{

SID = 1;

TenSV = "Nguyen Van Nam";

Khoa = "CNTT";

DiemTB = 7;

}

public Student(Student stu) //Constructor sao chep

{

SID = stu.SID;

TenSV = stu.TenSV;

Khoa = stu.Khoa;

DiemTB = stu.DiemTB;

}

//Constructor tham so

public Student(int id, string ten, string kh, float dtb)

{

SID = id;

TenSV = ten;

Khoa = kh;

DiemTB = dtb;

}

//Cac phuong thuc getter va setter cho tung thuoc tinh cua lop

public int GetStudentID()

{

return SID;

}

public void SetStudentID(int value)

{

SID = value;

}

public String GetName()

{

return TenSV;

}

public void SetName(String value)

{

TenSV = value;

}

public String GetFaculty()

{

return Khoa;

}

public void SetFaculty(String value)

{

Khoa = value;

}

public float GetMark()

{

return DiemTB;

}

public void SetMark(float value)

{

DiemTB = value;

}

//Phuong thuc hien thi du lieu

public void Show()

{

Console.WriteLine("MSSV:{0}", this.SID);

Console.WriteLine("Ten SV:{0}", this.TenSV);

Console.WriteLine("Khoa:{0}", this.Khoa);

Console.WriteLine("Diem TB:{0}", this.DiemTB);

}

}

class Tester

{

static Student Nhap1SV()

{

Student sv = new Student();

Console.Write("Nhap MaSV:");

sv.SetStudentID(int.Parse(Console.ReadLine()));

Console.Write("Ho ten SV:");

sv.SetName(Console.ReadLine());

Console.Write("Nhap khoa:");

sv.SetFaculty(Console.ReadLine());

Console.Write("Nhap Diem TB:");

sv.SetMark(float.Parse(Console.ReadLine()));

return sv;

}

static Student[] NhapDS(int n)

{

Student[] DSSV = new Student[n];

Console.WriteLine("\n ====NHAP DS SINH VIEN====");

for (int i = 0; i < n; i++)

{

Console.WriteLine("Nhap thong tin sinh vien thu {0}:", i + 1);

DSSV[i] = Nhap1SV();

}

return DSSV;

}

static void XuatDS(Student[] DSSV)

{

Console.WriteLine("\n ====XUAT DS SINH VIEN====");

foreach (Student sv in DSSV)

sv.Show();

}

public static void Main()

{

int n;

Console.Write("Nhap so luong SV:");

n = int.Parse(Console.ReadLine());

Student[] DSSV = NhapDS(n);

XuatDS(DSSV);

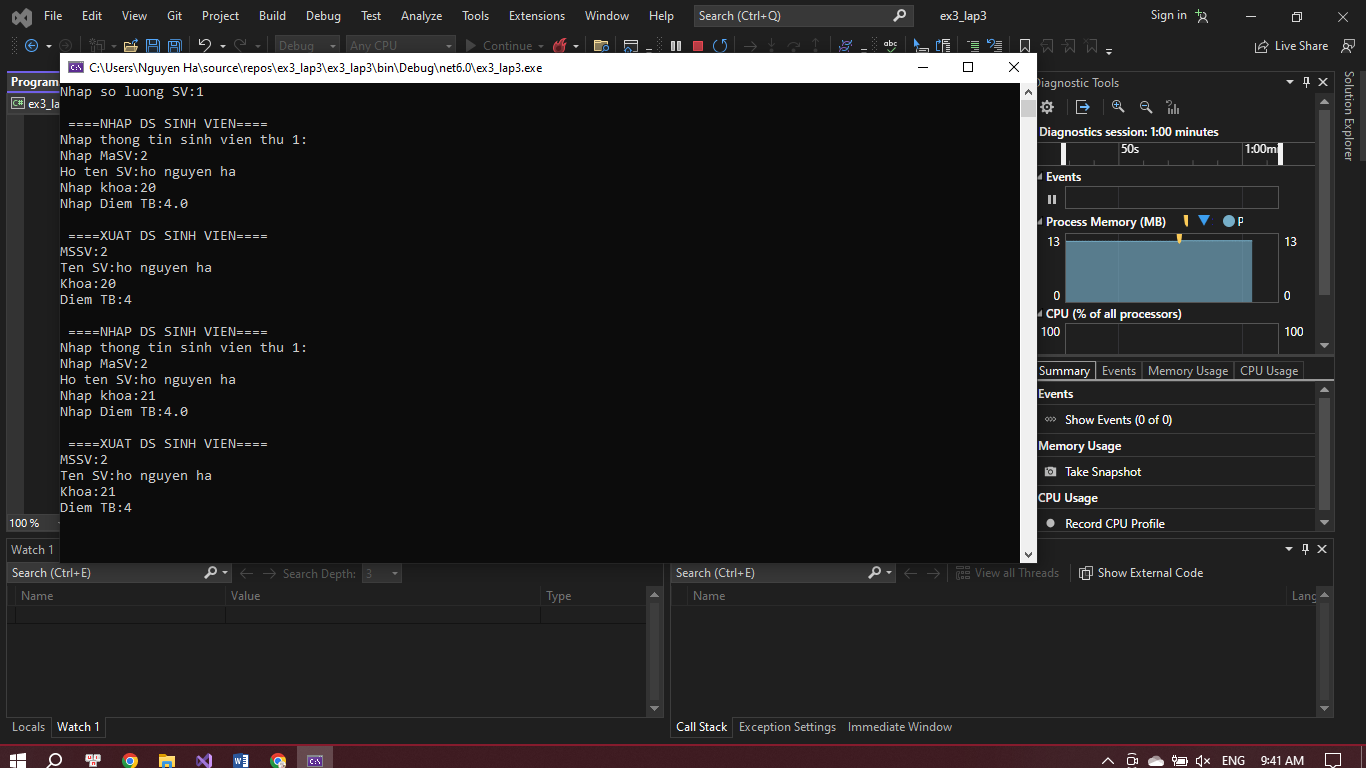
Console.ReadLine();

}

}

}

**Ex3:**



using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

// Khai báo không gian tên

namespace Lab01

{

// Định nghĩa lớp People

class People

{

// Khai báo các thuộc tính

protected int SID; // Mã sinh viên

protected string TenSV; // Tên sinh viên

// Phương thức khởi tạo mặc định

public People()

{

SID = 1;

TenSV = "Nguyen Van Nam";

}

// Phương thức khởi tạo sao chép

public People(People p)

{

SID = p.SID;

TenSV = p.TenSV;

}

// Phương thức khởi tạo với tham số

public People(int id, string ten)

{

SID = id;

TenSV = ten;

}

// Các phương thức getter và setter cho các thuộc tính

public int StudentID

{

get { return SID; }

set { SID = value; }

}

public String Name

{

get { return TenSV; }

set { TenSV = value; }

}

}

// Định nghĩa lớp Student kế thừa từ lớp People

class Student : People

{

// Khai báo các thuộc tính mới

private string Khoa; // Khoa học của sinh viên

private float DiemTB; // Điểm trung bình của sinh viên

// Phương thức khởi tạo mặc định

public Student() : base()

{

Khoa = "CNTT";

DiemTB = 7;

}

// Phương thức khởi tạo sao chép

public Student(Student stu) : base(stu)

{

Khoa = stu.Khoa;

DiemTB = stu.DiemTB;

}

// Phương thức khởi tạo với tham số

public Student(int id, string ten, string kh, float dtb) : base(id, ten)

{

Khoa = kh;

DiemTB = dtb;

}

// Các phương thức getter và setter cho các thuộc tính mới

public String Faculty

{

get { return Khoa; }

set { Khoa = value; }

}

public float Mark

{

get { return DiemTB; }

set { DiemTB = value; }

}

// Phương thức hiển thị thông tin sinh viên

public void Show()

{

Console.WriteLine("MSSV:{0}", this.SID);

Console.WriteLine("Ten SV:{0}", this.TenSV);

Console.WriteLine("Khoa:{0}", this.Khoa);

Console.WriteLine("Diem TB:{0}", this.DiemTB);

}

}

// Lớp Tester chứa các phương thức để nhập và xuất thông tin sinh viên từ console.

class Tester

{

static Student Nhap1SV()

{

Student sv = new Student();

Console.Write("Nhap MaSV:");

sv.StudentID = int.Parse(Console.ReadLine());

Console.Write("Ho ten SV:");

sv.Name = Console.ReadLine();

Console.Write("Nhap khoa:");

sv.Faculty = Console.ReadLine();

Console.Write("Nhap Diem TB:");

sv.Mark = float.Parse(Console.ReadLine());

return sv;

}

static List<Student> NhapDS(int n)

{

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

Console.WriteLine("\n ====NHAP DS SINH VIEN====");

for (int i = 0; i < n; i++)

{

Console.WriteLine("Nhap thong tin sinh vien thu {0}:", i + 1);

DSSV.Add(Nhap1SV());

}

return DSSV;

}

static void XuatDS(List<Student> DSSV)

{

Console.WriteLine("\n ====XUAT DS SINH VIEN====");

foreach (Student sv in DSSV)

sv.Show();

}

static ArrayList NhapDSArrayList(int n)

{

ArrayList DSSV = new ArrayList();

Console.WriteLine("\n ====NHAP DS SINH VIEN====");

for (int i = 0; i < n; i++)

{

Console.WriteLine("Nhap thong tin sinh vien thu {0}:", i + 1);

DSSV.Add(Nhap1SV());

}

return DSSV;

}

static void XuatDSArrayList(ArrayList DSSV)

{

Console.WriteLine("\n ====XUAT DS SINH VIEN====");

foreach (Student sv in DSSV)

sv.Show();

}

public static void Main()

{

int n;

Console.Write("Nhap so luong SV:");

n = int.Parse(Console.ReadLine());

List<Student> DSSVList = NhapDS(n);

XuatDS(DSSVList);

ArrayList DSSVArrayList = NhapDSArrayList(n);

XuatDSArrayList(DSSVArrayList);

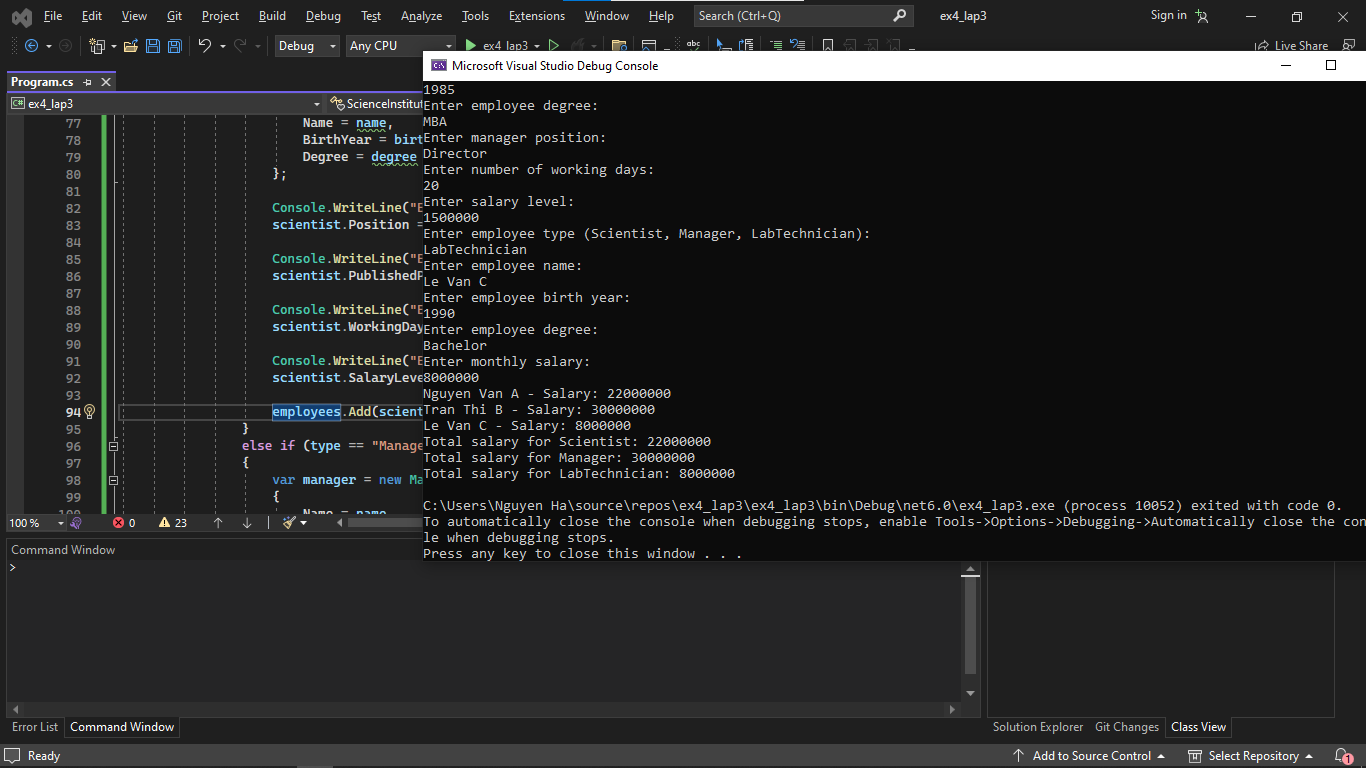
Console.ReadLine();

}

}

}

**Ex4:**



using System;

using System.Collections.Generic;

namespace ScienceInstituteManagement

{

// Lớp trừu tượng Employee đại diện cho một nhân viên

public abstract class Employee

{

// Tên của nhân viên

public string Name { get; set; }

// Năm sinh của nhân viên

public int BirthYear { get; set; }

// Bằng cấp của nhân viên

public string Degree { get; set; }

// Phương thức trừu tượng để tính lương của nhân viên

public abstract double CalculateSalary();

}

// Lớp Scientist kế thừa từ lớp Employee, đại diện cho một nhà khoa học

public class Scientist : Employee

{

// Chức vụ của nhà khoa học

public string Position { get; set; }

// Số lượng bài báo đã công bố của nhà khoa học

public int PublishedPapers { get; set; }

// Số ngày làm việc của nhà khoa học

public int WorkingDays { get; set; }

// Mức lương của nhà khoa học

public double SalaryLevel { get; set; }

// Triển khai phương thức CalculateSalary() để tính lương của nhà khoa học

public override double CalculateSalary()

{

return WorkingDays \* SalaryLevel;

}

}

// Lớp Manager kế thừa từ lớp Employee, đại diện cho một quản lý

public class Manager : Employee

{

// Chức vụ của quản lý

public string Position { get; set; }

// Số ngày làm việc của quản lý

public int WorkingDays { get; set; }

// Mức lương của quản lý

public double SalaryLevel { get; set; }

// Triển khai phương thức CalculateSalary() để tính lương của quản lý

public override double CalculateSalary()

{

return WorkingDays \* SalaryLevel;

}

}

// Lớp LabTechnician kế thừa từ lớp Employee, đại diện cho một kỹ thuật viên phòng thí nghiệm

public class LabTechnician : Employee

{

// Lương hàng tháng của kỹ thuật viên phòng thí nghiệm

public double MonthlySalary { get; set; }

// Triển khai phương thức CalculateSalary() để tính lương của kỹ thuật viên phòng thí nghiệm

public override double CalculateSalary()

{

return MonthlySalary;

}

}

class Program

{

static void Main(string[] args)

{

var employees = new List<Employee>();

Console.WriteLine("Enter number of employees: ");

int numEmployees = int.Parse(Console.ReadLine());

for (int i = 0; i < numEmployees; i++)

{

Console.WriteLine("Enter employee type (Scientist, Manager, LabTechnician): ");

string type = Console.ReadLine();

Console.WriteLine("Enter employee name: ");

string name = Console.ReadLine();

Console.WriteLine("Enter employee birth year: ");

int birthYear = int.Parse(Console.ReadLine());

Console.WriteLine("Enter employee degree: ");

string degree = Console.ReadLine();

if (type == "Scientist")

{

var scientist = new Scientist

{

Name = name,

BirthYear = birthYear,

Degree = degree

};

Console.WriteLine("Enter scientist position: ");

scientist.Position = Console.ReadLine();

Console.WriteLine("Enter number of published papers: ");

scientist.PublishedPapers = int.Parse(Console.ReadLine());

Console.WriteLine("Enter number of working days: ");

scientist.WorkingDays = int.Parse(Console.ReadLine());

Console.WriteLine("Enter salary level: ");

scientist.SalaryLevel = double.Parse(Console.ReadLine());

employees.Add(scientist);

}

else if (type == "Manager")

{

var manager = new Manager

{

Name = name,

BirthYear = birthYear,

Degree = degree

};

Console.WriteLine("Enter manager position: ");

manager.Position = Console.ReadLine();

Console.WriteLine("Enter number of working days: ");

manager.WorkingDays = int.Parse(Console.ReadLine());

Console.WriteLine("Enter salary level: ");

manager.SalaryLevel = double.Parse(Console.ReadLine());

employees.Add(manager);

}

else if (type == "LabTechnician")

{

var labTechnician = new LabTechnician

{

Name = name,

BirthYear = birthYear,

Degree = degree

};

Console.WriteLine("Enter monthly salary: ");

labTechnician.MonthlySalary = double.Parse(Console.ReadLine());

employees.Add(labTechnician);

}

}

}

}

}