Bài thực hành số 1

Ngôn ngữ C# - căn bản

Mục tiêu:

- Giúp sinh viên làm quen với ngôn ngữ C#: qua việc viết các ứng dụng **console** đơn giản, xây dựng các lớp, tạo đối tượng, truy xuất các phương thức, các câu lệnh…
- Làm quen với môi trường phát triển tích hợp VS .NET: các công cụ hỗ trợ soạn thảo mã nguồn, các công cụ biên dịch, debug…

Nội dung:

Tạo lớp Student có các dữ liệu và phương thức sau:

- SID (mã số sinh viên)
- Tên sinh viên
- Khoa
- Điểm TB
- Thêm các property cho các dữ liệu thành viên trên
- Viết các phương thức hiển thị thông tin của sinh viên.

Tạo lớp Tester, trong lớp này chỉ chứa duy nhất hàm Main(). Hàm cho phép người dùng nhập vào số **n** là số sinh viên, sau đó lần lượt tạo các đối tượng sinh viên và add vào danh sách sinh viên theo những thông tin do user nhập vào (dùng vòng lặp for). Cuối cùng xuất ra danh sách chi tiết thông tin sinh viên.

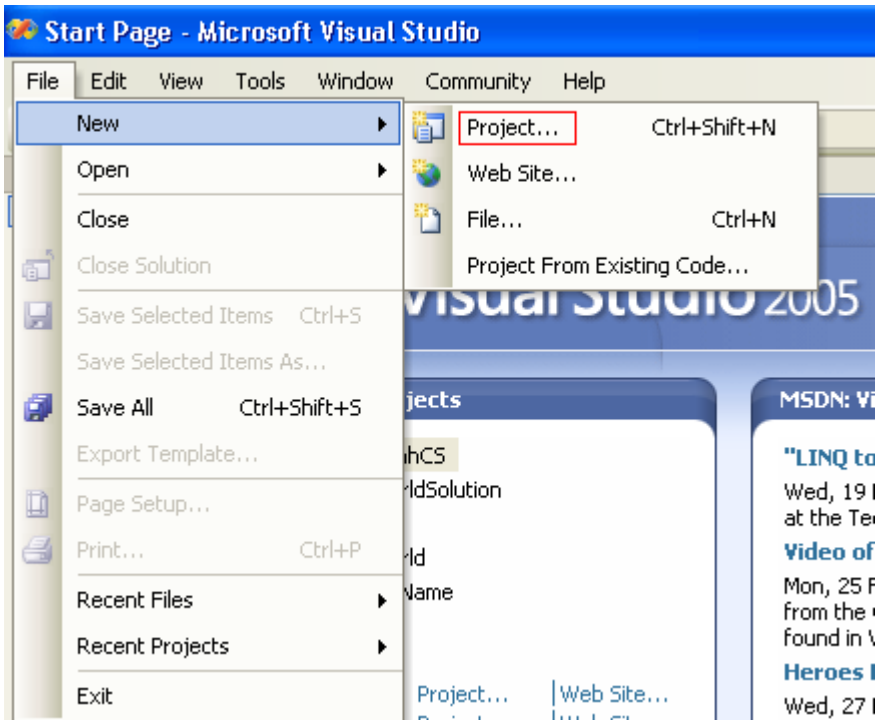
Yêu cầu:

- Sinh viên xây dựng chương trình theo nội dung mô tả bên trên.

- Compile & Build chương trình.
- Run chương trình ở hai chế độ debug và không debug.
- 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.

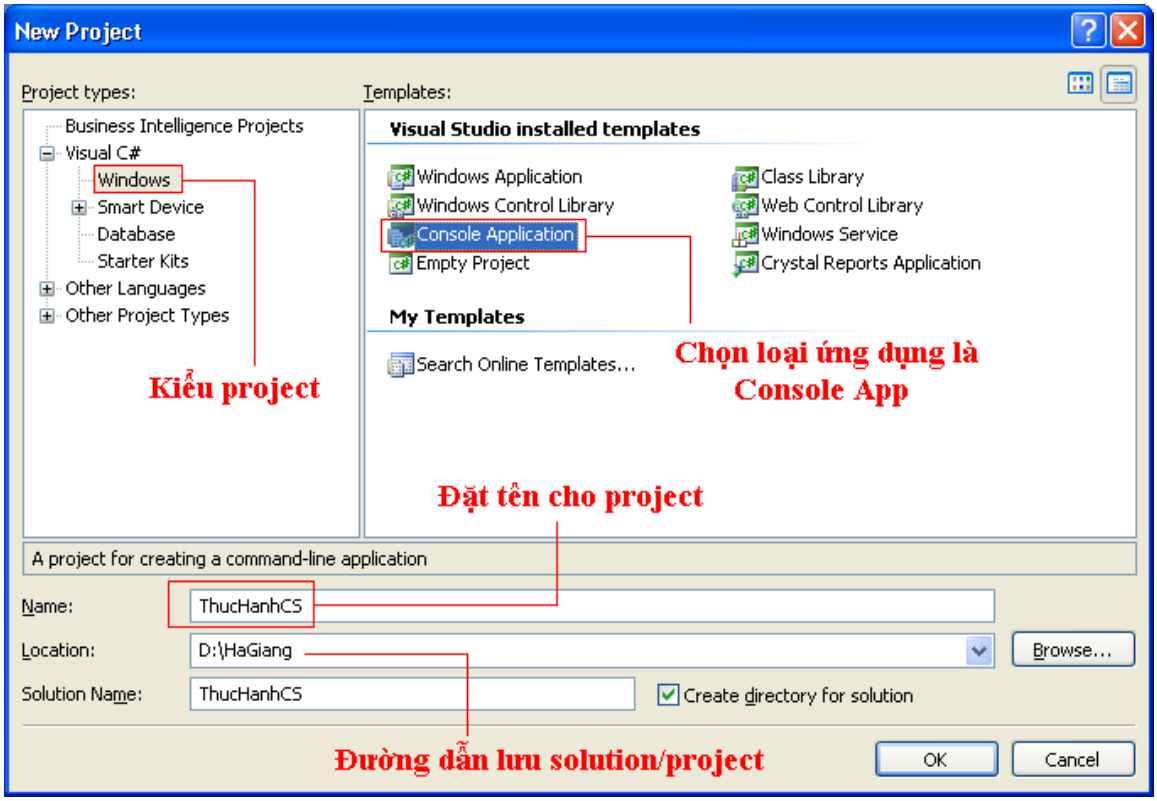
Hướng dẫn:

1. Tạo project trong VS .NET:
    1. Trong menu File chọn New → Poject hoặc nhấn tổ hợp phím (Ctrl+Shift+N), xuất hiện cửa sổ New Project.



Hình 1: Màn hình tạo project

2. Trong cửa sổ **New Project**: chọn
    i. Project type là Visual C# - Windows
    ii. Chọn templates là Console Application
    iii. Nhập tên project vào phần Name: **ThucHanhCS**
    iv. Khai báo đường dẫn lưu trữ trong Location…
    v. Khai báo tên Solution Name…

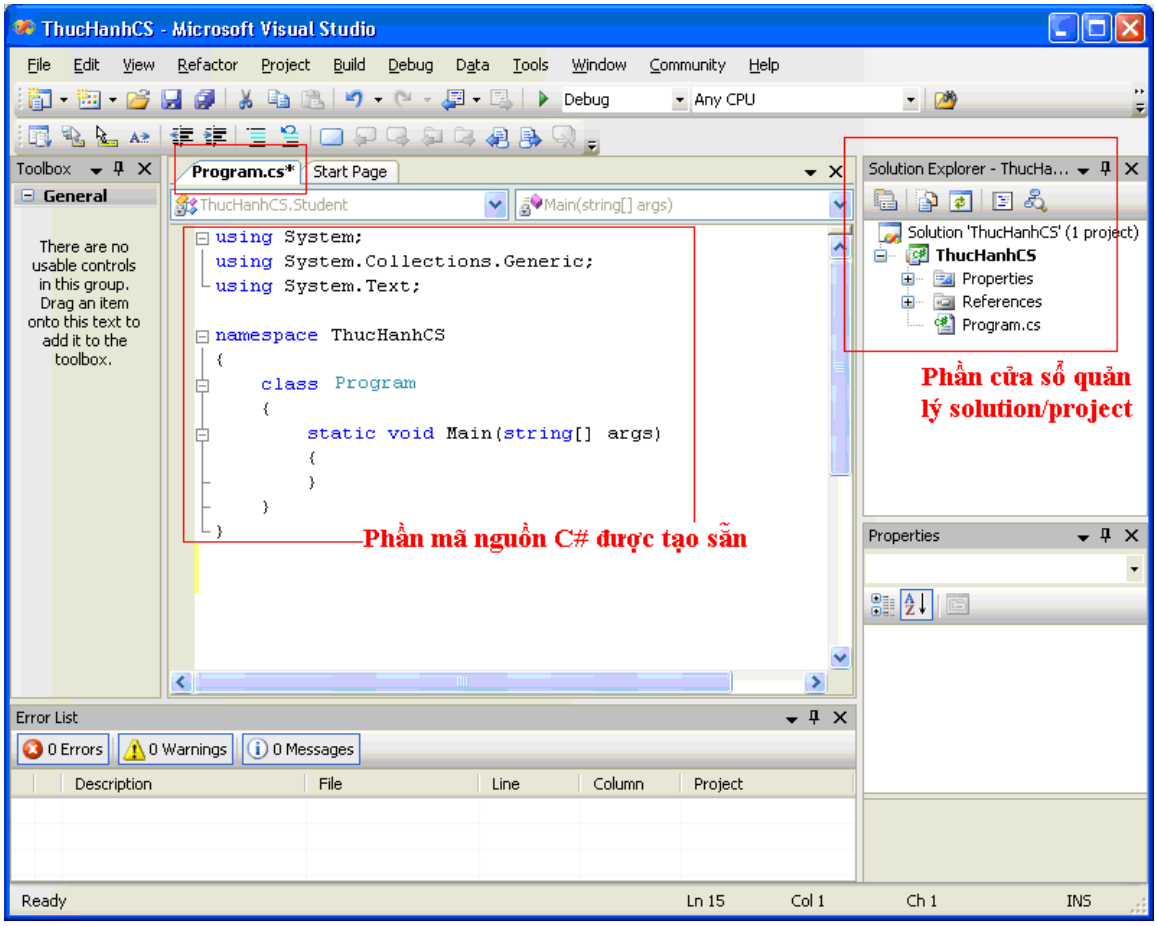


Hình 2: Màn hình chọn loại project.

3. Nhấn phím **OK** để kết thúc quá trình tạo project, kết quả chúng ta được một khung sườn ứng dụng console như minh họa bên dưới

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ThucHanhCS
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

Toàn bộ màn hình làm việc của Project vừa tạo trong VS .NET như sau:



Hình 3: Toàn bộ project ThucHanhCS được VS khởi tạo.

2. Xây dựng các lớp theo yêu cầu của bài tập

    1. Xóa lớp Program mặc định do Wizard tạo ra

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ThucHanhCS
{

}
```

    2. Tạo lớp Student bên trong namespace ThucHanhCS

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ThucHanhCS
{
    public class Student
    {

    }
}
```

3. Thêm các trường dữ liệu cho lớp Student

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ThucHanhCS
{
    public class Student
    {
        // các dữ liệu thành viên
        private int SID;
        private string Ten;
        private string Khoa;
        private float Diem;


    }
}
```

4. Tạo các hàm khởi dựng: hàm thứ nhất mặc định không tham số, hàm thứ hai sao chép và hàm thứ ba có tham số.

```
public class Student
{
    // các dữ liệu thành viên
    private int SID;
    private string Ten;
    private string Khoa;
    private float Diem;

    public Student()
    {
        SID = 1;
        Ten = "Nguyen Ha Nam";
        Khoa = "CNTT";
        Diem = 10;
    }
    public Student(Student stu)
    {
        SID = stu.SID;
        Ten = stu.Ten;
        Khoa = stu.Khoa;
        Diem = stu.Diem;
    }
    public Student(int id, string ten, string khoa, float diem)
    {
        SID = id;
        Ten = ten;
        Khoa = khoa;
        Diem = diem;
    }
}
```

dữ liệu thành viên

Constructor ko tham số

Hàm constructor sao chép

Hàm constructor có tham số

5. Tạo các **property** cho từng dữ liệu thành viên của lớp. Đây là hình thức truy xuất dữ liệu thành viên của lớp trong C# (cách truyền thống là dùng accessor gồm phương thức Getter và Setter của lớp để truy xuất các field dạng private).

```
// Danh sách các properties
public int STUDENTID
{
    get
    {
        return SID;
    }
    set
    {
        SID = value;
    }
}
public string NAME
{
    get { return Ten; }
    set { Ten = value;}
}
public string FACULTY
{
    get { return Khoa; }
    set { Khoa = value; }
}
public float MARK
{
    get { return Diem; }
    set { Diem = value; }
}
```

Property "STUDENTID" đại diện cho dữ liệu SID

Lấy dữ liệu

Gán dữ liệu

Tên property

6. Viết phương thức Show thể hiện thông tin của lớp Student

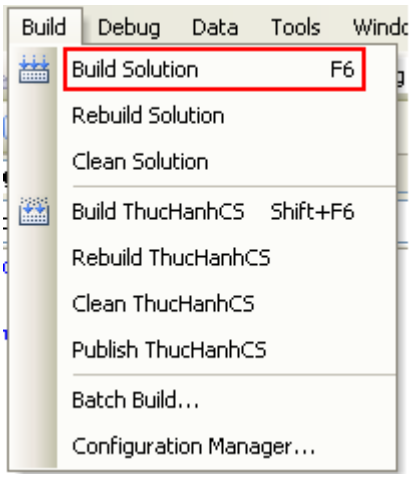
```
// Phương thức

public void Show()
{
    Console.WriteLine("MSSV: {0}",this.SID);
    Console.WriteLine("Ten SV: {0}", this.NAME);
    Console.WriteLine("Khoa: {0}", this.FACULTY);
    Console.WriteLine("Diem: {0}",this.MARK);
}
```

7. Kết thúc việc tạo lớp Student, phần tiếp theo chúng ta sẽ tạo lớp Tester. Lớp Tester chỉ chứa duy nhất hàm Main. Hàm Main có các chức năng sau:
   - Đọc vào n là số lượng sinh viên
   - Tạo mảng chứa n sinh viên
   - Lần lượt nhập thông tin của từng sinh viên vào danh sách

- Hiển thị thông tin của từng sinh viên.

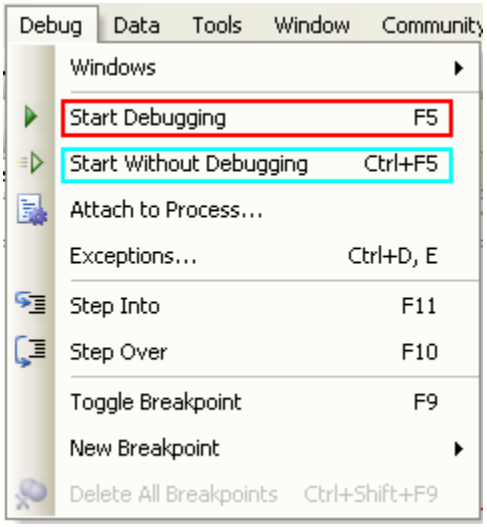
```
public static void Main()
{
    Student[] StudentList;
    int n;

    Console.Write("Nhap so luong sinh vien: ");
    n = int.Parse(Console.ReadLine());

    StudentList = new Student[n]; // tạo mảng có n phần tử

    for (int i = 0; i < n; i++)
    {
        StudentList[i] = new Student(); // tạo đối tượng sinh viên
        Console.Write("Nhap ten sv {0}: ",i+1);
        StudentList[i].NAME = Console.ReadLine();
        Console.Write("Nhap MS: ");
        StudentList[i].STUDENTID = int.Parse(Console.ReadLine());
        Console.Write("Khoa: ");
        StudentList[i].FACULTY = Console.ReadLine();
        Console.Write("Diem: ");
        StudentList[i].MARK = int.Parse(Console.ReadLine());
    }
    // Xuất danh sách sinh viên
    Console.WriteLine("*******DANH SACH SINH VIEN*******");
    foreach (Student st in StudentList)
        st.Show();

    Console.ReadLine();
```

8. Biên dịch và chạy chương trình:
   - Chức năng Build Solution: F6



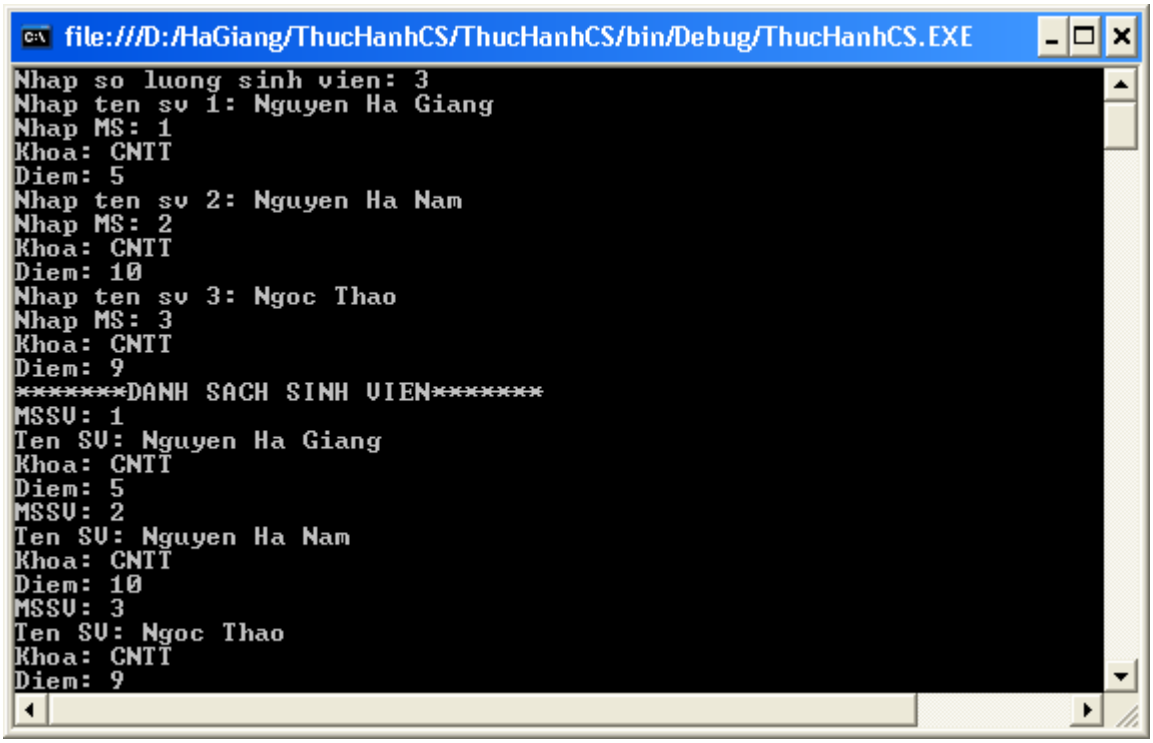
Hình 4: Chức năng Build

- Chức năng Run with Debug: F5



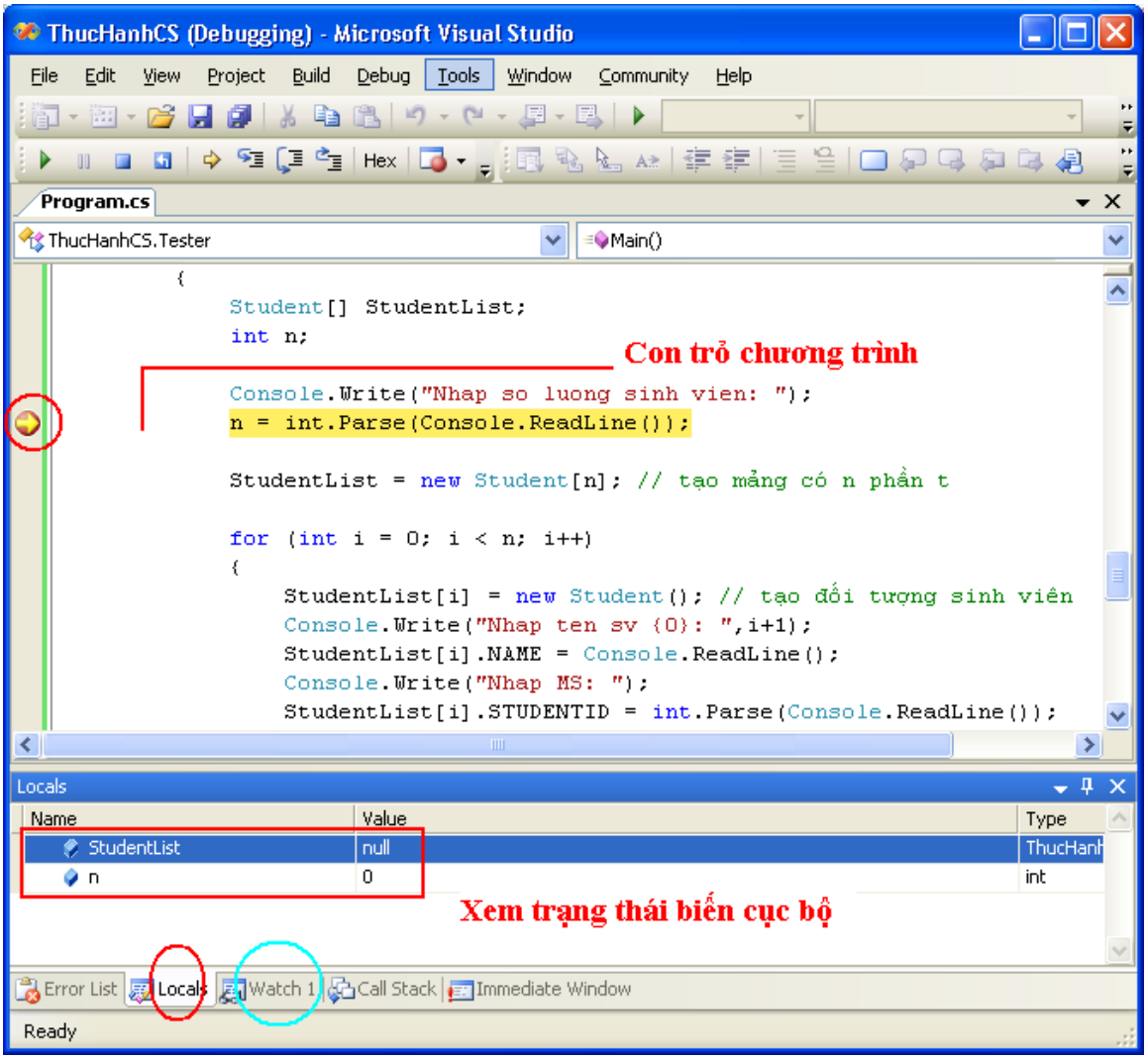
Hình 5: Chức năng Debug/ without Debug

- Chức năng Run without Debug: Ctrl + F5



Hình 6: Màn hình chương trình khi thực hiện

9. Sinh viên chạy debug chương trình: sử dụng các chức năng breakpoint, xem giá trị của các biến trong cửa sổ Locals hoặc nhập các biến vào cửa sổ Watch1 để xem giá trị hiện tại của nó.



Hình 7: Màn hình chương trình chạy debug dừng tại một breakpoint.

Cách thức chạy từng bước chương trình trong chế độ Debug:

- Cách thứ nhất chèn cách breakpoint vào một dòng lệnh nào đó: trong màn hình soạn thảo, di chuyển con trỏ văn bản tới dòng cần dừng nhấn <**F9**> hay kích chuột vào lề trái của dòng đó sẽ xuất hiện ký hiệu breakpoint

  Khi chạy debug thì chọn <**F5**> chương trình sẽ thực hiện và dừng tại breakpoint, muốn chạy tiếp thì tiếp tục nhấn <**F5**>

  Để remove breakpoint thì di chuyển con trỏ văn bản vào dòng đó và nhấn <**F9**>

- Cách thứ hai chạy từng dòng lệnh, bắt đầu từ hàm Main():

  Nhấn <**F10**>: chương trình sẽ chạy debug vào lệnh đầu tiên của hàm Main

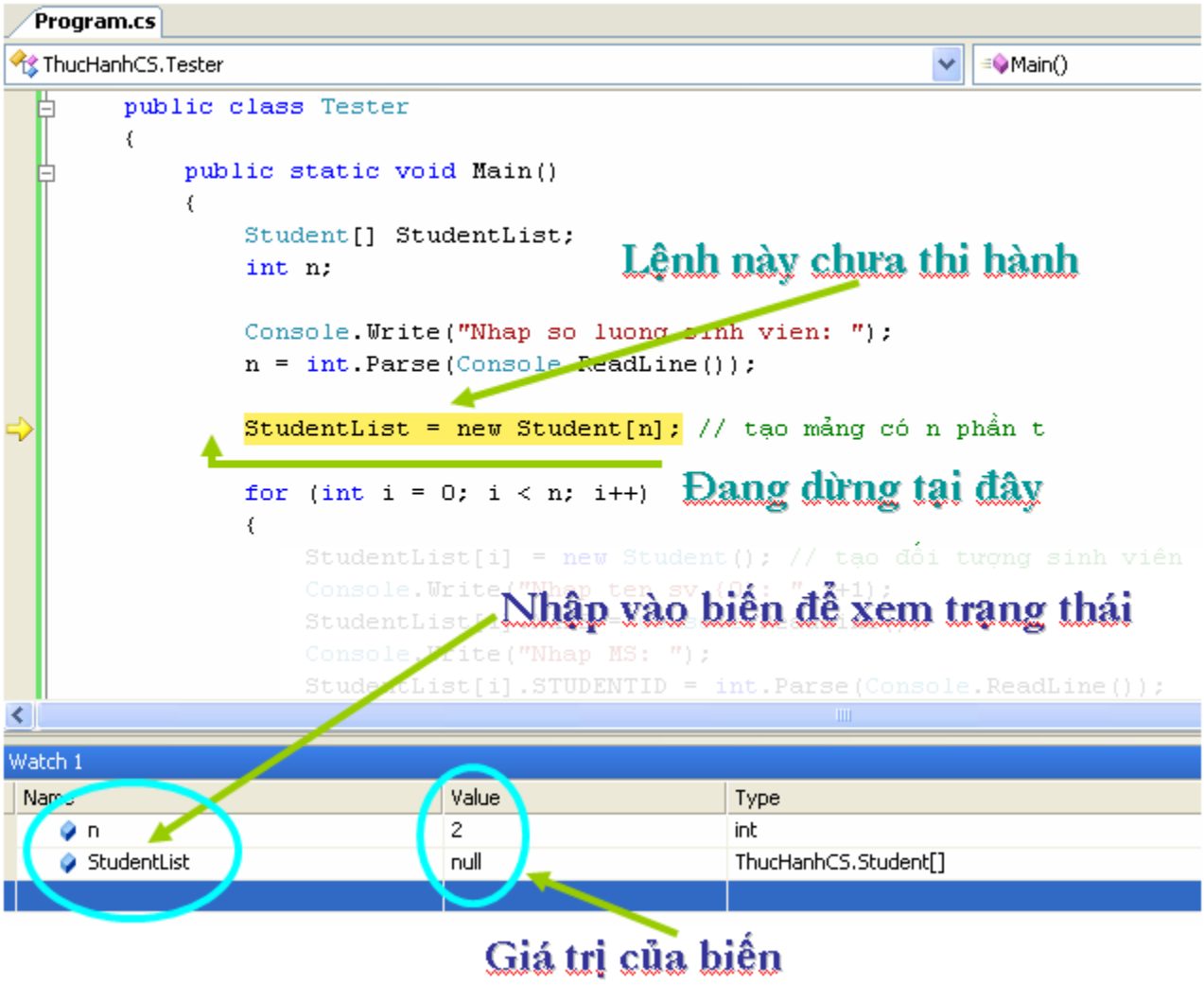
  Chạy từng dòng lệnh thì nhấn <**F10**>

Vào trong thân của một lời gọi hàm <**F11**>

Thoát ra khỏi hàm nào đó thì nhấn <**Shift + F11**>

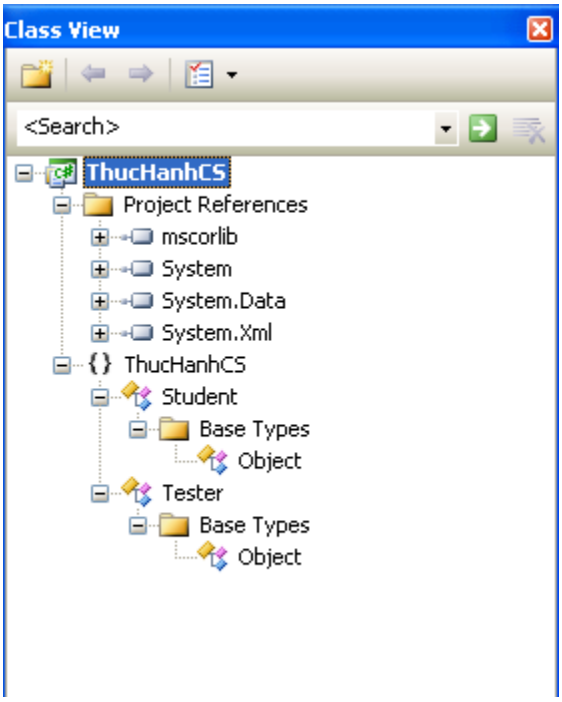
Để kết thúc debug ở bất cứ nơi nào thì nhấn: <**Shift + F5**>

Trong quá trình debug có thể kết hợp với của sổ Locals hay Watch1 để xem thông tin chi tiết của các biến, đối tượng trong chương trình.



Hình 8: Màn hình trạng thái debug chương trình

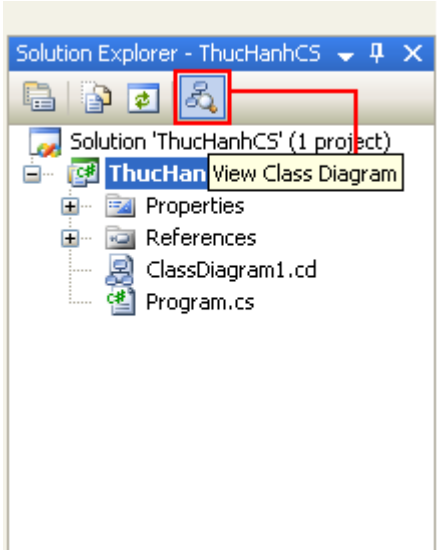
10. Sử dụng cửa sổ **Class View** (Ctrl + W, C) để quản lý các lớp có trong chương trình



Hình 9: Cửa sổ Class View của project

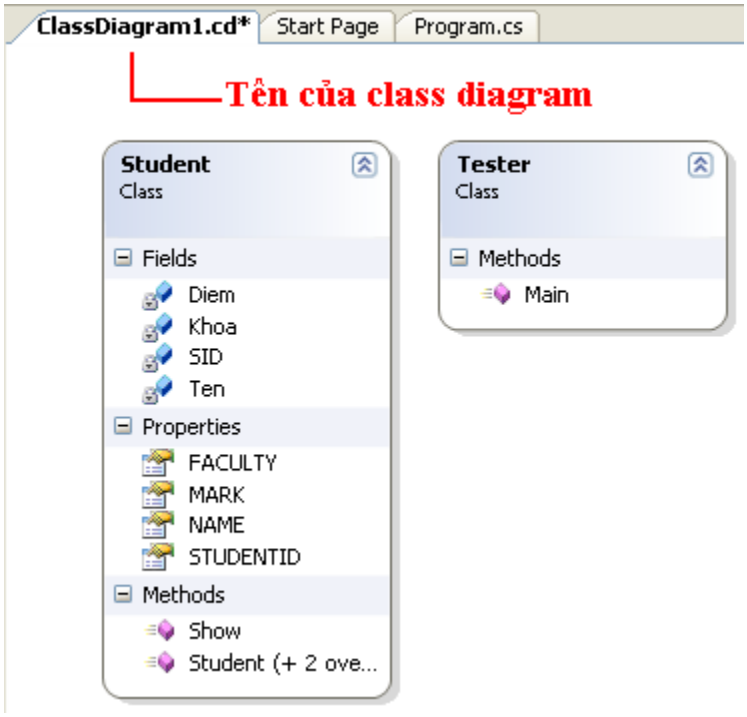
11. Sử dụng chức năng **View Class Diagram** để xem các lớp có trong chương trình:

- Trong cửa sổ Solution Explorer chọn button **View Class Diagram**



Hình 10: Chọn chức năng View Class Diagram

- Sau khi chọn chức năng này, VS.NET 2005 sẽ khởi tạo ra class diagram và lưu vào một file có phần mở rộng là ***.cd**



Hình 11: Class diagram do VS.NET phát sinh theo source code chương trình

Lưu ý: Sinh viên có thể bổ sung các thuộc tính như địa chỉ, số điện thoại, hoặc chi tiết các cột điểm…