**TRƯỜNG ĐẠI HỌC ĐỒNG THÁP**

**KHOA SƯ PHẠM TOÁN - TIN**

**BÁO CÁO BÀI TẬP LỚN MÔN HỌC**

**PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG**

Lớp: DHSPTIN20 – Chuyên ngành: SP TIN HỌC

Đề tài:

**QUẢN LÝ CỬA HÀNG XE MÁY**

**GVHD: Ths.Trần Kim Hương**

**Thành viên nhóm:**

**5420440160– Nguyễn Thị Hồng**

**5420440165-Huỳnh Thị Hồng Ngân**

**5420440175-Hồ Thanh Toàn**

**5420440178-Phan Thị Cẩm Tú**

**Đồng Tháp – 5/7/2022**

# LỜI NÓI ĐẦU

Chúng tôi xin cam đoan Bài tập lớn “Quản lý cửa hàng xe máy” là công trình nghiên cứu của bản thân. Những phần sử dụng tài liệu tham khảo trong Bài tập lớn đã được nêu rõ trong phần tài liệu tham khảo. Các số liệu, kết quả trình bày trong Bài tập lớn là hoàn toàn trung thực, nếu sai Chúng tôi xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của bộ môn và nhà trường đề ra.

*Đồng Tháp, ngày … tháng … năm…..*

Sinh viên

                                                  …………………………………..

Mục lục

[LỜI NÓI ĐẦU 2](#_Toc107606168)

[PHẦN MỘT: MỞ ĐẦU 4](#_Toc107606169)

[PHẦN HAI: NỘI DUNG 5](#_Toc107606170)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 5](#_Toc107606171)

[CHƯƠNG 2. PHÂN TÍCH PHẦN MỀM HƯỚNG ĐỐI TƯỢNG 24](#_Toc107606172)

[CHƯƠNG 3. XÂY DỰNG PHẦN MỀM HƯỚNG ĐỐI TƯỢNG 29](#_Toc107606173)

[BÀI TẬP CÁ NHÂN – SV1: Nguyễn Thị Hồng 42](#_Toc107606174)

[BÀI TẬP CÁ NHÂN – SV2: Huỳnh Hồng Ngân 45](#_Toc107606175)

[BÀI TẬP CÁ NHÂN – SV3: Hồ Thanh Toàn 48](#_Toc107606176)

[BÀI TẬP CÁ NHÂN – SV4: Phan Thị Cẩm Tú 50](#_Toc107606177)

[PHẦN BA: KẾT LUẬN 53](#_Toc107606178)

[1. HƯỚNG DẪN SỬ DỤNG 54](#_Toc107606179)

[2. BẢNG PHÂN CÔNG CÔNG VIỆC (Bài tập nhóm) 55](#_Toc107606180)

[TÀI LIỆU THAM KHẢO 56](#_Toc107606181)

# PHẦN MỘT: MỞ ĐẦU

1. **Đặt vấn đề**

Như chúng ta đã thấy, thời đại của công nghiệp 4.0, việc ứng dụng công nghệ thông tin vào việc quản lý đã trở lên vô cùng phổ biến. Chính vì vậy, để đáp ứng nhu cầu đó, chúng tôi quyết định xây dựng ứng dụng Quản lý cửa hàng xe máy để cho phép người dùng dễ dàng quản lý, bán hàng, thống kê một cách thuận tiện và đơn giản.

1. **Ý nghĩa lý thuyết, thực tiễn của đề tài**

Phần mềm giúp người dùng dễ dàng quản lý cũng như mua bán xe một cách dễ dàng, thống kê doanh thu cũng như sản phẩm bán chạy

1. **Mục tiêu của đề tài**

Xây dựng ứng dụng Quản lý cửa hàng xe máy

1. **Phạm vi của đề tài**

* Phạm vi không gian: Một số cửa hàng bán xe máy
* Phạm vi thời gian: từ tháng 5-6 năm 2022

1. **Các phương pháp nghiên cứu**

* Khảo sát yêu cầu của người dùng
* Tìm hiểu thông tin khác liên quan tới phần mềm

1. **Kế hoạch thực hiện đề tài**

Phần mềm sẽ có những chức năng sau:

* Quản lý tài khoản, nhân viên
* Quản lý bán hàng
* Quản lý nhập hàng
* Quản lý đại lý
* Quản lý xe
* Quản lý hóa đơn

# PHẦN HAI: NỘI DUNG

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1. **Lập trình hướng đối tượng**

**Lập trình hướng đối tượng (OOP) là gì?**

Lập trình hướng đối tượng (OOP) là một kỹ thuật lập trình cho phép [lập trình viên](https://topdev.vn/blog/lap-trinh-vien/) tạo ra các đối tượng trong code trừu tượng hóa các đối tượng.

Đối tượng là những sự vật, sự việc mà nó có những tính chất, đặc tính, hành động giống nhau và ta gom góp lại thành đối tượng giống trong thực tế cuộc sống. Khi lập trình OOP, chúng ta sẽ định nghĩa các lớp (class) để gom (mô hình) các đối tượng thực tế.

**Đối tượng**

Một đối tượng bao gồm 2 thông tin: thuộc tính và phương thức.

 Thuộc tính chính là những thông tin, đặc điểm của đối tượng. Ví dụ: con người có các đặc tính như mắt, mũi, tay, chân…

 Phương thức là những thao tác, hành động mà đối tượng đó có thể thực hiện. Ví dụ: một người sẽ có thể thực hiện hành động nói, đi, ăn, uống, . . .

**Lớp**

Một lớp là một kiểu dữ liệu bao gồm các thuộc tính và các phương thức được định nghĩa từ trước. Đây là sự trừu tượng hóa của đối tượng. Khác với kiểu dữ liệu thông thường, một lớp là một đơn vị (trừu tượng) bao gồm sự kết hợp giữa các phương thức và các thuộc tính. Hiểu nôm na hơn là các đối tượng có các đặc tính tương tự nhau được gom lại thành một lớp đối tượng.

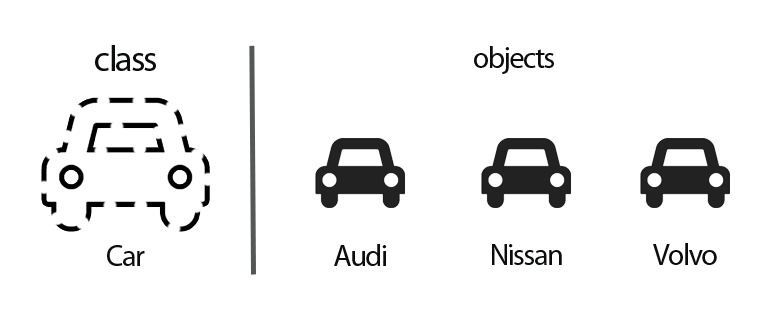
**Sự khác nhau giữa đối tượng và lớp**

Lớp bạn có thể hiểu nó như là khuôn mẫu, đối tượng là một thực thể thể hiện dựa trên khuôn mẫu đó. Ví dụ: Ta nói về loài chó, bạn có thể hiểu nó là class (lớp) chó có:

Các thông tin, đặc điểm: 4 chân, 2 mắt, có đuôi, có chiều cao, có cân nặng, màu lông…

Các hành động như: sủa, đi, ăn, ngủ…

Đối tượng thì chính là con chó Phú Quốc ta đang nuôi trong nhà cũng mang đặc tính của lớp chó.



**Các nguyên lý cơ bản của OOP**

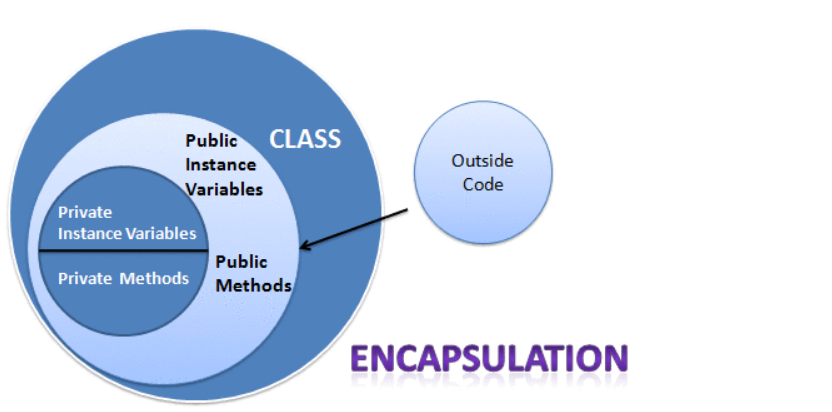
**Tính đóng gói (Encapsulation)**

Note:

Các dữ liệu và phương thức có liên quan với nhau được đóng gói thành các lớp để tiện cho việc quản lý và sử dụng. Tức là mỗi lớp được xây dựng để thực hiện một nhóm chức năng đặc trưng của riêng lớp đó.

 Ngoài ra, đóng gói còn để che giấu một số thông tin và chi tiết cài đặt nội bộ để bên ngoài không thể nhìn thấy.

Các thuộc tính và phương thức của đối tượng cần cho việc giải quyết bài toán đã được chọn ra sẽ được đóng gói vào một kiểu dữ liệu gọi là class. Nhằm che dấu thông tin khỏi các đối tượng bên ngoài, thuộc tính nào thì phương thức nấy. Tức là code bên ngoài không trực tiếp nhìn thấy và thay đổi trạng thái của đối tượng đó.



Nói chung trạng thái đối tượng không hợp lệ thường do: chưa được kiểm tra tính hợp lệ, các bước thực hiện không đúng trình tự hoặc bị bỏ qua nên trong OOP có một quy tắc quan trọng cần nhớ đó là phải luôn khai báo các trạng thái bên trong của đối tượng là private và chỉ cho truy cập qua các public/protected method/property.

Ví dụ ta thấy một viên thuốc chữa cảm. Chúng ta chỉ biết nó chữa cảm sổ mũi nhức đầu và một số thành phần chính, còn cụ thể bên trong nó có những hoạt chất gì thì hoàn toàn không biết.

**Tính kế thừa (Inheritance)**

Nó cho phép xây dựng một lớp mới dựa trên các định nghĩa của lớp đã có. Có nghĩa là lớp cha có thể chia sẽ dữ liệu và phương thức cho các lớp con. Các lớp con khỏi phải định nghĩa lại, ngoài ra có thể mở rộng các thành phần kế thừa và bổ sung thêm các thành phần mới. Tái sử dụng mã nguồn 1 cách tối ưu, tận dụng được mã nguồn. Một số loại kế loại kế thừa thường gặp: đơn kế thừa, đa kế thừa, kế thừa đa cấp, kế thừa thứ bậc.

Khi bắt đầu xây dựng ứng dụng chúng ta sẽ bắt đầu thiết kế định nghĩa các lớp trước. Thông thường một số lớp có quan hệ với những lớp khác, chúng có những đặc tính giống nhau.

VD: 2 lớp Android, iPhone

Mỗi lớp đều đại diện cho một loại smartphone khác nhau nhưng lại có những thuộc tính giống nhau như gọi điện, nhắn tin, chụp hình. Thay vì sao chép những thuộc tính này, ta nên đặt chúng vào một lớp chung gọi là lớp cha. Chúng ta có thể định nghĩa lớp cha – trong trường hợp này là Smartphone và có những lớp con kế thừa từ nó, tạo ra một mối quan hệ cha/con.

**Tính đa hình (Polymorphism)**

Tính đa hình là một hành động có thể được thực hiện bằng nhiều cách khác nhau. Đây lại là một tính chất có thể nói là chứa đựng hầu hết sức mạnh của lập trình hướng đối tượng.

Hiểu một cách đơn giản hơn: Đa hình là khái niệm mà hai hoặc nhiều lớp có những phương thức giống nhau nhưng có thể thực thi theo những cách thức khác nhau.

Ví dụ như ở phần trên, mỗi một smartphone kế thừa từ lớp cha Smartphone nhưng có thể lưu trữ dữ liệu trên cloud theo những cách khác nhau. Android lưu trữ bằng Google Drive iPhone lưu trên iCloud.

Một ví dụ về đa hình trong thực tế. Ta có 2 con vật: chó, mèo. Cả 2 con vật này đều là lớp động vật. Nhưng khi ta bảo cả 2 động vật kêu thì con chó sẽ kêu gâu gâu, con mèo sẽ kêu meo meo.

Vậy trong ví dụ chó, mèo xem như là các đối tượng. 2 con vật có thể hiểu cùng kêu nhưng theo các cách khác nhau.

**Tính trừu tượng (Abstraction)**

Trừu tượng có nghĩ là tổng quát hóa một cái gì đó lên, không cần chú ý chi tiết bên trong. Nó không màng đến chi tiết bên trong là gì và người ta vẫn hiểu nó mỗi khi nghe về nó.

Ví dụ: Bạn chạy xe tay ga thì có hành động là tăng ga để tăng tốc, thì chức năng tăng ga là đại diện cho trừu tượng (abstraction). Người dùng chỉ cần biết là tăng ga thì xe tăng tốc, không cần biết bên trong nó làm thế nào.

Ở đây trong lập trình OOP, tính trừu tượng nghĩa là chọn ra các thuộc tính, phương thức của đối tượng cần cho việc giải quyết bài toán đang lập trình. Vì một đối tượng có rất nhiều thuộc tính phương thức, nhưng với bài toán cụ thể không nhất thiết phải chọn tất cả.

Ví dụ: Bài toán quản lý sinh viên chúng ta chỉ cần quản lý các thông tin như

Họ tên

Ngày sinh

Giới tính

…

Điểm thi

mà lại không cần quản lý thêm các thông tin:

Màu tóc

Sở thích

Chiều cao

Tại vì chúng thực sự không cần thiết.

Các ưu điểm của lập trình hướng đối tượng

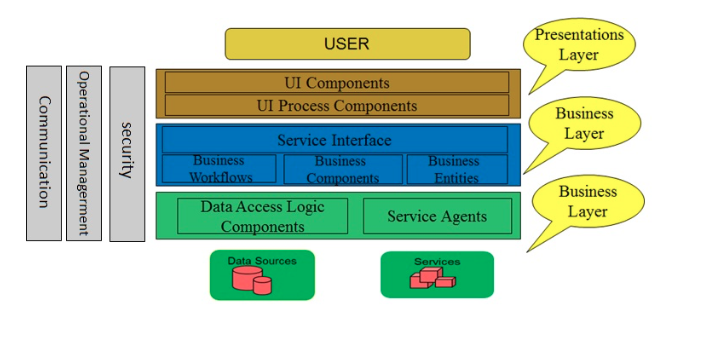
Dựa trên nguyên lý kế thừa, trong quá trình mô tả các lớp có thể loại bỏ những chương trình bị lặp, dư. Và có thể mở rộng khả năng sử dụng các lớp mà không cần thực hiện lại. Tối ưu và tái sử dụng code hiệu quả.

Đảm bảo rút ngắn thời gian xây dựng hệ thống và tăng năng suất thực hiện.

Sự xuất hiện của 2 khái niệm mới là lớp và đối tượng chính là đặc trưng của phương pháp lập trình hướng đối tượng. Nó đã giải quyết được các khuyết điểm của phương pháp lập trình hướng cấu trúc để lại. Ngoài ra 2 khái niệm này đã giúp biểu diễn tốt hơn thế giới thực trên máy tính.

1. **Mô hình 3 lớp**

**Giới thiệu mô hình 3-layer ( 3 lớp)**



Mô hình 3-layer gồm có 3 phần chính :

–**Presentation Layer (GUI) :** Lớp này có nhiệm vụ chính giao tiếp với người dùng. Nó gồm các thành phần giao diện ( win form, web form,…) và thực hiện các công việc như nhập liệu, hiển thị dữ liêu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business Logic Layer (BLL).

– **Business Logic Layer (BLL) :** Layer này phân ra 2 thành nhiệm vụ :

Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL.

Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer.

– **Data Access Layer (DAL) :** Lớp này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu ( tìm kiếm, thêm, xóa, sửa,…).

Ưu điểm

Việc phân chia thành từng lớp giúp cho code được tường minh hơn. Nhờ vào việc chia ra từng lớp đảm nhận các chức năng khác nhau và riêng biệt như giao diện, xử lý, truy vấn thay vì để tất cả lại một chỗ. Nhằm giảm sự kết dính.

Dễ bảo trì khi được phân chia, thì một thành phần của hệ thống sẽ dễ thay đổi. Việc thay đổi này có thể được cô lập trong 1 lớp, hoặc ảnh hưởng đến lớp gần nhất mà không ảnh hưởng đến cả chương trình.

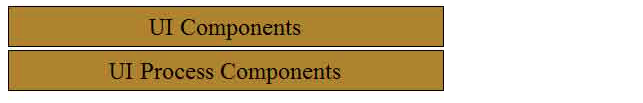
Dễ phát triển, tái sử dụng: khi chúng ta muốn thêm một chức năng nào đó thì việc lập trình theo một mô hình sẽ dễ dàng hơn vì chúng ta đã có chuẩn để tuân theo. Và việc sử dụng lại  khi có sự thay đổi giữa hai môi trường ( Winform sang Webfrom ) thì chỉ việc thay đổi lại lớp GUI.

Dễ bàn giao. Nếu mọi người đều theo một quy chuẩn đã được định sẵn, thì công việc bàn giao, tương tác với nhau sẽ dễ dàng hơn và tiết kiệm được nhiều thời gian.

Dễ phân phối khối lượng công việc. Mỗi một nhóm, một bộ phận sẽ nhận một nhiệm vụ trong mô hình 3 lớp. Việc phân chia rõ ràng như thế sẽ giúp các lập trình viên kiểm soát được khối lượng công việc của mình.

Phân tích chi tiết từng layer trong mô hình 3 lớp.

**1. Presentation Layer (GUI):**



Có hai thành phần chính sau đây với những tác vụ cụ thể :

UI Components : gồm các thành phần tạo nên giao diện của ứng dụng (GUI). Chúng chịu trách nhiệm thu nhận và hiển thị dữ liệu cho người dùng… Ví dụ : textbox, button, combobox, …

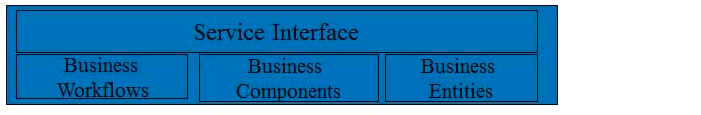
UI Process Components : là thành phần chịu trách nhiệm quản lý các quá trình chuyển đổi giữa các UI… Ví dụ : Sắp xếp quá trình kiểm tra thông tin khách hàng:

1.Hiển thị màn hình tra cứu ID

2.Hiển thị màn hình thông tin chi tiết khách hàng tương ứng

3.Hiển thị màn hình liên lạc với khách hàng.

2. Bussiness Layer (BLL) :



Lớp này gồm 4 thành phần:

Service Interface : là thành phần giao diện lập trình mà lớp này cung cấp cho lớp Presentation sử dụng.

Bussiness Workflows : chịu trách nhiệm xác định và điều phối các quy trình nghiệp vụ gồm nhiều bước và kéo dài. Những quy trình này phải được sắp xếp và thực hiện theo một thứ tự chính xác.

Ví dụ : Thực hiện mua một đơn hàng trên tiki qua nhiều bước : kiểm tra gói hàng còn không?, tính tổng chi phí, cho phép giao dịch và sắp xếp việc giao hàng.

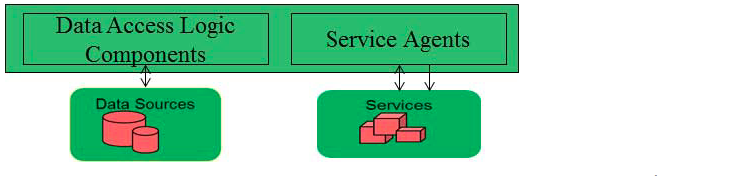
Bussiness Components : chịu trách nhiệm kiểm tra các quy tắc nghiệp vụ, ràng buộc logic và thực hiện các công việc . Các thành phần này cũng thực hiện các dịch vụ mà Service Interface cung cấp và Business Workflows sẽ sử dụng nó.

Ví dụ : Tiếp tục ví dụ ở trên. Bạn sẽ cần một Bussiness Component để kiểm tra gói hàng có khả dụng không ? hay một component để tính tổng chi phí,…

Bussiness Entities : thường được sử dụng như Data Transfer Objects ( DTO ) . Bạn có thể sử dụng để truyền dữ liệu giữa các lớp (Presentation và Data Layer). Chúng thường là cấu trúc dữ liệu ( DataSets, XML,… ) hay các lớp đối tượng đã được tùy chỉnh.

Ví dụ : tạo 1 class Student lưu trữ các dữ liệu về tên, ngày sinh, ID, lớp.

**3. Data Layer (DAL) :**

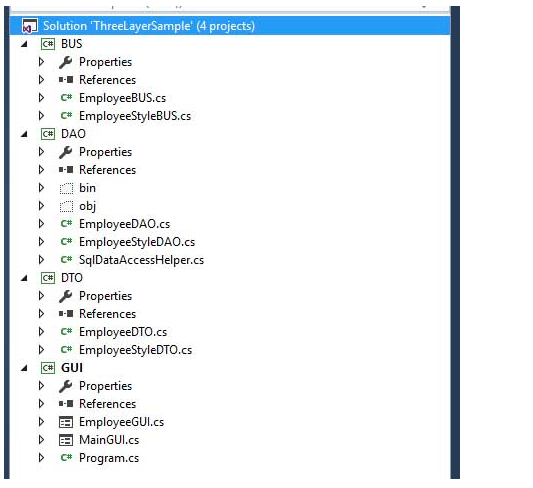


Data Access Logic Components : chịu trách nhiệm chính lưu trữ và truy xuất dữ liệu từ các nguồn dữ liệu ([Data Sources](https://topdev.vn/blog/dung-gi-de-luu-tru-data-thay-the-local-storage/)) như XML, file system,… Hơn nữa còn tạo thuận lợi cho việc dễ cấu hình và bảo trì.

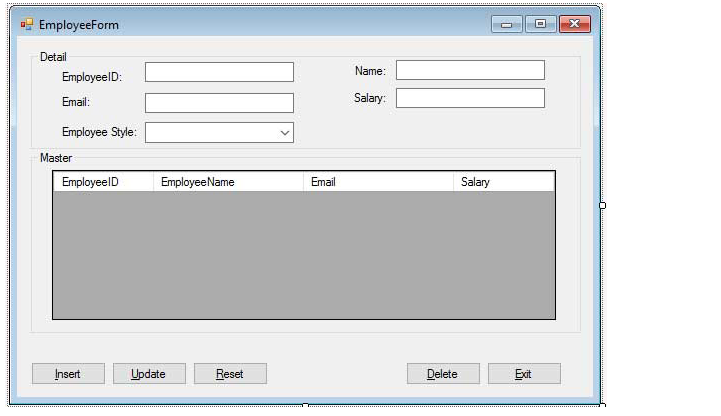
 Service Agents : giúp bạn gọi và tương tác với các dịch vụ từ bên ngoài một cách dễ dàng và đơn giản.

Cấu trúc mô hình 3 lớp

Để hiểu rõ hơn về cấu trúc và cách xây dựng của mô hình 3 lớp, chúng ta cùng tham khảo một ví dụ về mô hình quản lí công nhân gồm các lớp BUS, DAO, GUI. (Các đoạn code sẽ bị lược bỏ bớt )



Đầu tiên là GUI gồm các button insert, update, reset ,delete ,exit .Người dùng sẽ giao tiếp với màn hình giao diện này



Lớp DTO, đây không phải là layer, đây chỉ là 1 gói dữ liệu đươc trao đổi giữa các lớp. Gói dữ liệu này được xây dựng dưới dạng lớp đối tượng. Mỗi một công nhân sẽ mang những thuộc tính sau:

namespace DTO

{

public class EmployeeDTO

{

#region Atrributes

private String \_employeeID;

private String \_name;

private String \_email;

private float \_salary;

private int \_employeeStyle;

#endregion

//.....

Các nghiệp vụ xử lý chính sẽ được đặt ở lớp BUS (hay là BLL)  gồm các nghiệp vụ insert, update, delete, retrieve

namespace BUS

{

public class EmployeeBUS

{

#region 1. Inserting

public static bool InsertEmployee(EmployeeDTO emp)

{

if (EmployeeDAO.CheckEmployeeByID(emp.EmployeeID)==true

&amp;&amp;EmployeeStyleDAO.CheckEmployeeStyleByID(emp.EmployeeStyle)==false)

{

return false;

}

return EmployeeDAO.InsertEmployee(emp);

}

#endregionhttps://techtalk.vn/wp-admin/post-new.php#

//2. Updating

//3. Deleting

//4. Retrieving

}

}

Và cuối cùng là lớp DAO ( hay là DAL ). Truy  vấn đến cơ sở dữ liệu

{

public class EmployeeDAO

{

#region 1. Inserting

public static bool InsertEmployee(EmployeeDTO emp)

{

bool result=false;

try

{

// Create List Sql Parameter

List sqlParams = new List();

sqlParams.Add(new SqlParameter("@EmployeeID", emp.EmployeeID));

sqlParams.Add(new SqlParameter("@Name", emp.Name));

sqlParams.Add(new SqlParameter("@Email", emp.Email));

sqlParams.Add(new SqlParameter("@Salary", emp.Salary));

sqlParams.Add(new SqlParameter("@EmployeeStyle", emp.EmployeeStyle));

// Call Store Procedure

int n = SqlDataAccessHelper.ExecuteNoneQuery("spInsertEmployee", sqlParams);

if (n == 1)

result = true;

}

catch (Exception ex)

{

throw ex;

}

return result;

}

#endregion

// 2. Updating

// 3. Deleting

// 4. Retrieving

1. **Vài nét về LINQ**

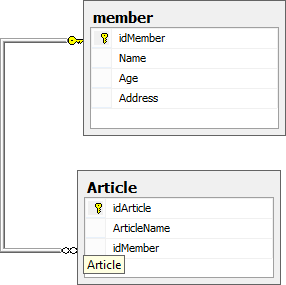
**LinQ to SQL là gì?**

LinQ có tên gọi là Language Integrated Query, một thư viện cung cấp cho ngôn ngữ lập trình C# và Visual Basic. Thư viện này cung cấp khả năng truy vấn database ngay trên ngôn ngữ lập trình. Khi sử dụng, LinQ sẽ mô phỏng lại cơ sở dữ liệu và lưu trong trương trình, sau đó mỗi entity sẽ được xây dựng thành mỗi class như mô hình MVC, từ đó có thể truy xuất database thông qua những class đó.

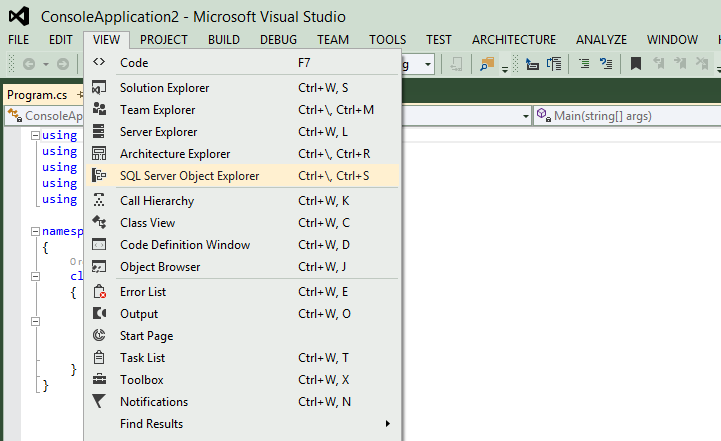
LinQ to SQL phù hợp với những dự án phần mềm như quản lý bán hàng, quản lý của hàng, nhân viên vì cung cấp khả năng tạo một database ngay trong SQL Server thông qua mô hình database mà LinQ to SQL được người lập trình cung cấp trong quá trình xây dựng sản phẩm. Rút ngắn đáng kể thời gian cho lập trình viên và dễ dàng cho người sử dụng cài đặt.

**Cài đặt**

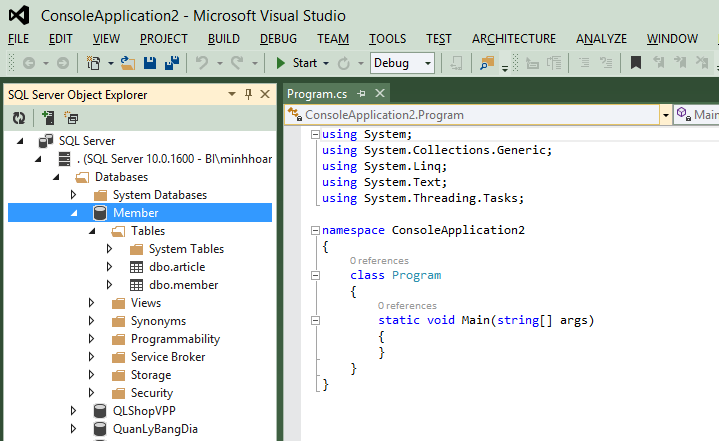
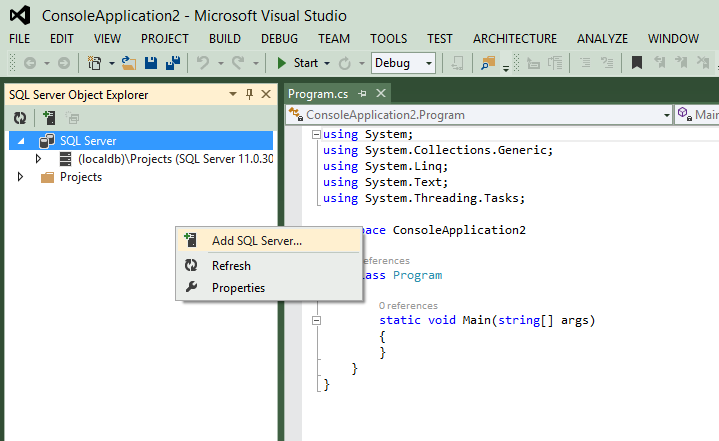
Bước 1. Tạo một database có tên Member, trong database này gồm có 2 entity member và article.



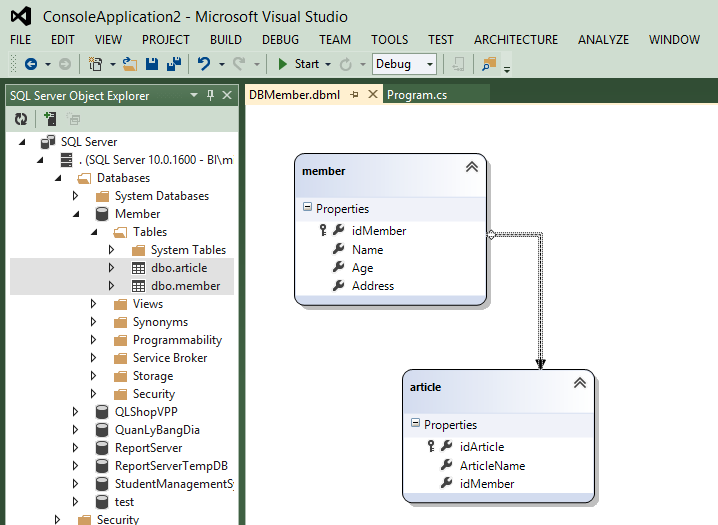
Bước 2. Tạo một project C# → vào View chọn SQL Server Object Explorer



Bước 3. Chọn Add SQL Server và connect vào SQL Server trên máy, kết quả được một list các database.



Bước 4. Chọn Add → New Item → chọn LINQ to SQL Classes và đặt lại tên (trong ví dụ này đặt tên cho file là DBMember.dbml), một file có đuôi dbml được tạo ra. Chọn tất cả các table trong database muốn sử dụng kéo vào trong file dbml này. Kết quả như sau.



Tạo xong LinQ to SQL, các class có cùng tên với mỗi entity cũng đã được tạo ra.

Các thao tác sử dụng LinQ to SQL

Khởi tạo class DBMemberDataContext, mọi việc truy xuất cơ sở dữ liệu đều thông qua class này.

Select

DBMemberDataContext db = new DBMemberDataContext();

var name = from table in db.members

where table.idMember == 0

select table.Name;

foreach(var a in name)

{

Console.WriteLine(a);

}

Tạo một một biến tên table và biến này chứa dựng tất cả dữ liệu có trong table member.

Điều kiện là tìm tất cả những những member nào có idMember bằng 0, có thể lấy theo điều kiện khác tùy loại điều kiện có số lượng kết quả trả về khác nhau. Và cuối cùng là chọn field muốn trả về.

Vì kết quả trả về là mảng nên dùng foreach để lấy tất cả dữ liệu trong đó.

Trong trường hợp trên chỉ có thể lấy được dữ liệu duy nhất đó là field name.

DBMemberDataContext db = new DBMemberDataContext(); // khởi tạo

var name = from table in db.members

where table.idMember == 0

select table;

foreach(var a in name)

{

Console.WriteLine(a.Name);

Console.WriteLine(a.Address);

}

Thay vì chỉ lấy được một field có thể lấy toàn bộ dữ liệu trong table thông qua điều kiện cho trước và sau đó truy xuất các field mong muốn.

Insert

DBMemberDataContext db = new DBMemberDataContext();

member mem = new member();

mem.idMember = 1;

mem.Name = "stdio";

mem.Age = 22;

mem.Address = "456 xyz";

db.members.InsertOnSubmit(mem);

db.SubmitChanges();

Đoạn code trên dùng để insert một dữ liệu vào database, trong trường hợp này xem nó như một đối tượng.

Trước tiên tạo ra một đối tượng member và nhập dữ liệu vào các thuộc tính của nó.

Tiếp theo sử dụng db.members.InsertOnSubmit(mem); để thêm đối tượng này vào nhưng đối tượng này hiện tại chỉ được thêm tạm thời vào DBMemberDataContext nhưng chưa được cập nhật vào databate của SQL Server.

Vì vậy thêm dòng code db.SubmitChanges(); để đồng bộ tất cả các dữ liệu của project và SQL Server.

Update

DBMemberDataContext db = new DBMemberDataContext();

var members = from table in db.members

where table.idMember == 0

select table;

foreach (var mem in members)

{

mem.Address = "Tân Bình";

}

db.SubmitChanges();

Đoạn code này sẽ truy vấn đến database để tìm những member nào có idMember bằng 0 và sau đó thay đổi giá trị trong field Address, cuối cùng là đồng bộ lại dữ liệu.

Delete

DBMemberDataContext db = new DBMemberDataContext();

var members = from table in db.members

where table.idMember == 0

select table;

foreach (var mem in members)

{

db.members.DeleteOnSubmit(mem);

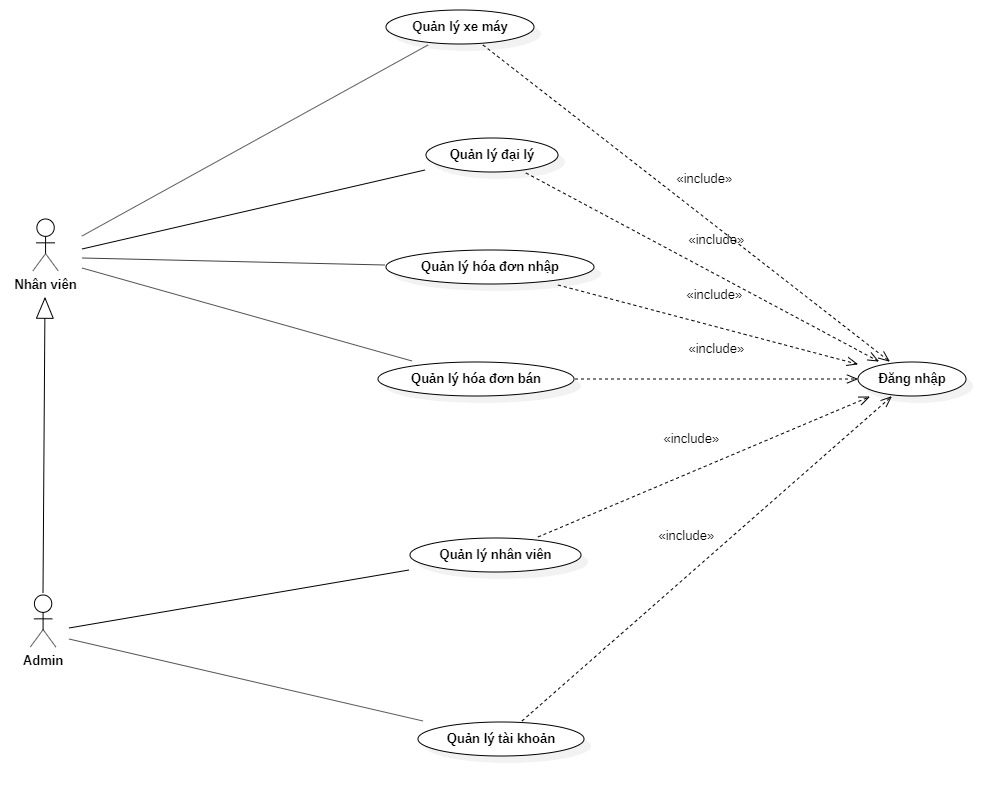
}

db.SubmitChanges();

Ví dụ này thực hiện việc delete một đối tượng có trong members. Cũng như update và insert sau khi dòng code db.members.DeleteOnSubmit(mem); được thực thi thì mới chỉ xóa được trên đối tượng trên DBMemberDataContext.

CHƯƠNG 2. PHÂN TÍCH PHẦN MỀM HƯỚNG ĐỐI TƯỢNG

1. **Mô tả bài toán**
2. **Phân tích phần mềm hướng đối tượng**
3. Sơ đồ Use-case



1. Đặc tả Use-case
   1. Đặc tả use-case

Kịch bản ca sử dụng QuanLyBan

**Mô tả tóm tắt**

Tên ca sử dụng: QuanLyBanSanPham

 Mục đích: Cho phép admin lập hóa đơn bán hàng và xử lý đơn và lưu trữ hóa đơn

 Tóm lược: được bắt đầu khi có khách hàng mua

 Lập hóa đơn: chức năng lập hóa đơn bán cho khách hàng bằng cách nhập mã sản phẩm, số lượng và giá tiền.

 Khi khách hàng vào giỏ hàng và thanh toán hóa đơn sẽ được lưu.

**Tác nhân: Admin**

Mô tả kịch bản:

 Điều kiện đầu vào: thông tin sản phẩm.

 Kịch bản chính: lập hóa đơn bán khách hàng đặt mua sản phẩm hệ thống sẽ tự đưa ra hóa đơn đặt hàng.

**Kịch bản ca sử dụng QuanLySanPham**

**Tác nhân: Admin.**

Mô tả: Đây là chức năng giành cho người quản lí.

Điều kiện cần: Đăng nhập vào hệ thống quản lí

Điều kiện đủ:

**Dòng sự kiện chính**

- Thêm sản phẩm

+ Admin chức năng thêm sản phẩm

+ Xuất hiện giao diện và admin nhập các thông tin của sản phẩm.

+ Click nút “Thêm”.

                    + Hệ thống kiểm tra dữ liệu đầu vào và thực hiện chức năng thêm

- Xóa sản phẩm

+ Admin chọn chức năng xóa ở sản phẩm cần xóa.

           + Xuất hiện giao diện và admin “Click” vào chức năng “Xóa ”.

                    + Hệ thống kiểm tra và thực hiện chức năng.

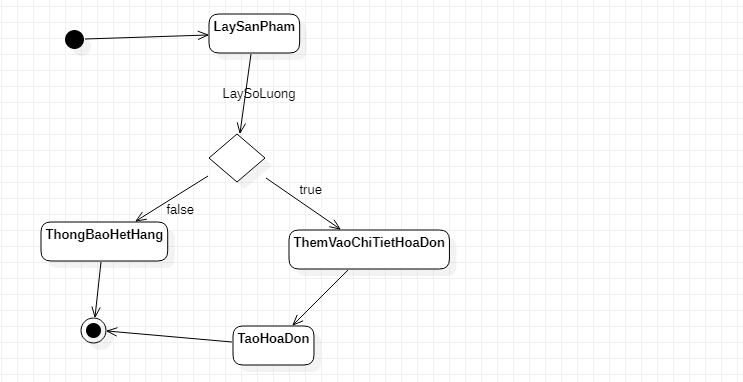
                    - Sửa sản phẩm

                     + Admin chọn chức năng sửa ở sản phẩm cần sửa.

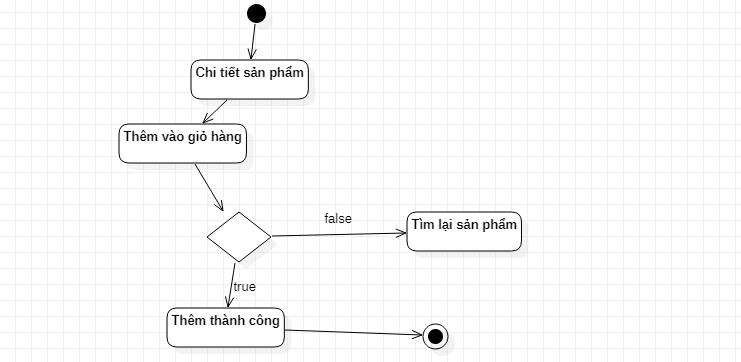
            +  Xuất hiện giao diện và admin “Click” vào chức năng “Sửa”.

            +  Hệ thống kiểm tra dữ liệu đầu vào và thực hiện chức năng sửa

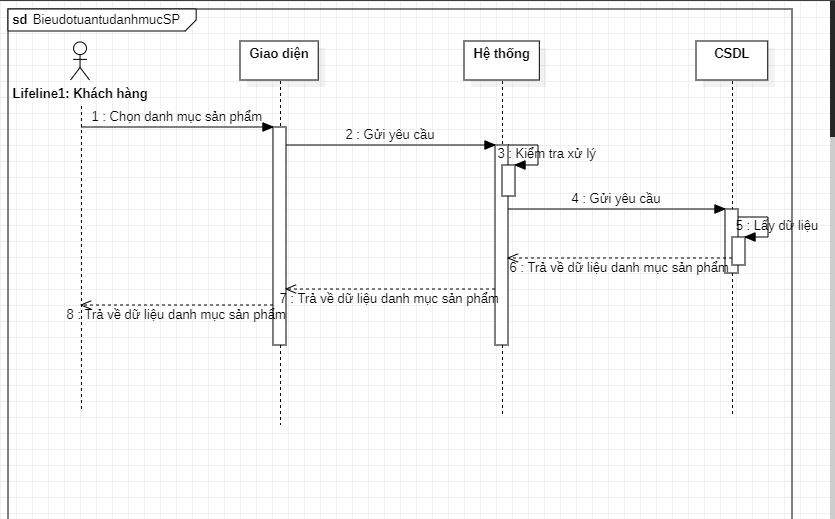
* 1. Sơ đồ hoạt động (activity diagram)

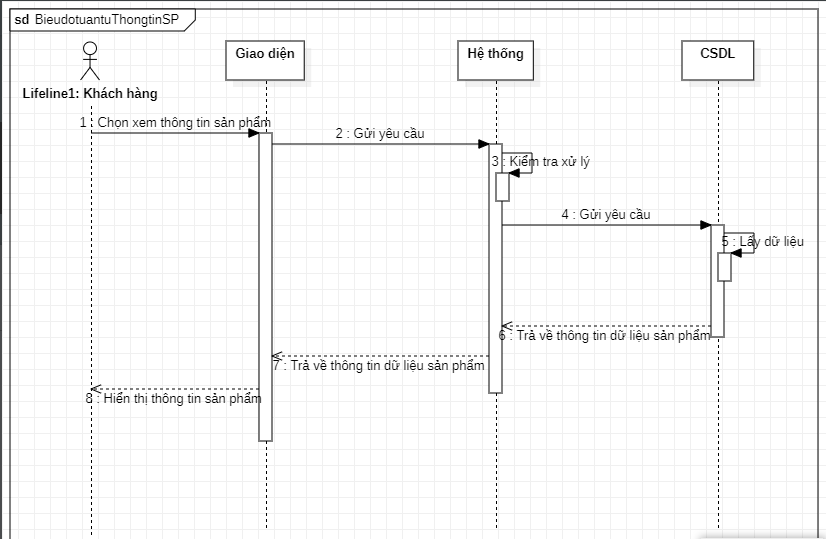


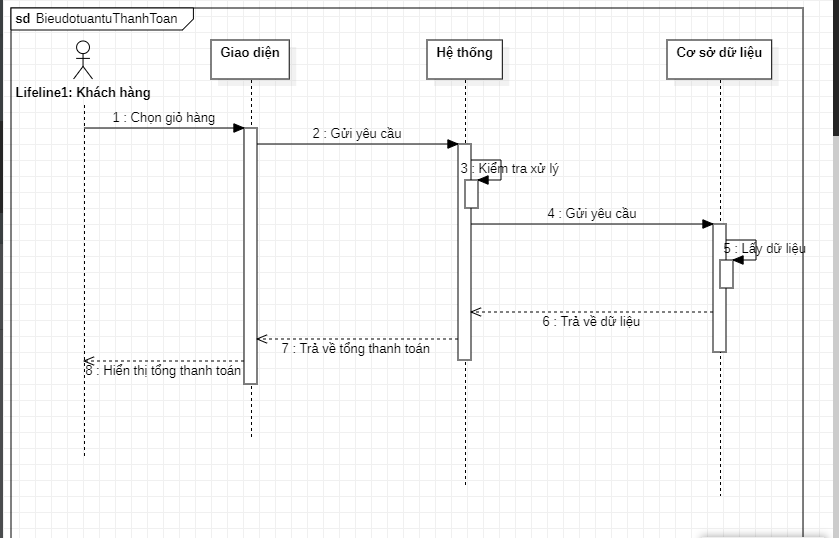


**

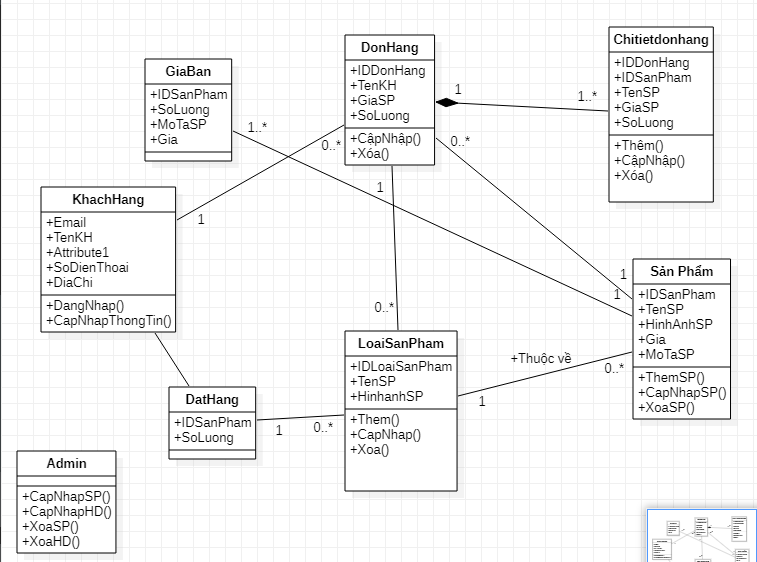
* 1. Sơ đồ tuần tự (sequence diagram)





**

1. Sơ đồ lớp (Class diagram)

**

CHƯƠNG 3. XÂY DỰNG PHẦN MỀM HƯỚNG ĐỐI TƯỢNG

1. **Cách tổ chức dữ liệu**
   1. Thiết kế dữ liệu vật lý – PDM
2. Bảng NhanVien:Lưu thông tin nhân viên

* *Bảng lưu trữ dữ liệu*

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã NV** | **Tên NV** | **SĐT** | **Địa chỉ** |
| *1* | *Nguyễn Văn Xuân* | *028343298* | *Bình Phước* |
| *2* | *Trần Văn Lan* | *023948348* | *Bình Thuận* |
| *3* | *Hoàng Xuân Thạch* | *023478789* | *Đồng Tháp* |

* *Bảng cấu trúc*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| *1* | *MaNV* | *Int* | *Primary key* | *Mã nhân viên* |
| *2* | *TenNV* | *Nvarchar(50)* | *Not null* | *Tên nhân viên* |
| *3* | *SDT* | *Char(10)* | *Not null* | *Số điện thoại* |
| *4* | *DiaChi* | *Nvarchar(50)* | *Not null* | *Địa chỉ* |

1. Bảng Tài khoản: Lưu thông tin tài khoản

* *Bảng lưu trữ dữ liệu*

|  |  |  |
| --- | --- | --- |
| **Tên đăng nhập** | **Mật khẩu** | **Loại tài khoản** |
| *Admin* | *Admin* | *Admin* |
| *Nv1* | *123* | *NhanVien* |
| *Nv2* | *123* | *NhanVien* |

* *Bảng cấu trúc*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| *1* | *TenTK* | *Varchar(20)* | *Primary key* | *Tên đăng nhập* |
| *2* | *MatKhau* | *Nvarchar(50)* | *Not null* | *Mật khẩu* |
| *3* | *LoaiTK* | *Char(10)* | *Not null* | *Loại tài khoản* |

1. Bảng Loại sản phẩm: Lưu thông tin loại xe

* *Bảng lưu trữ dữ liệu*

|  |  |
| --- | --- |
| **Mã loại** | **Tên loại** |
| *1* | *Xe côn tay* |
| *2* | *Xe ga* |
| *3* | *Xe số* |

* *Bảng cấu trúc*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| *1* | *Maloai* | *Int* | *Primary key* | *Mã loại* |
| *2* | *TenLoai* | *Nvarchar(50)* | *Not null* | *Tên loại* |

1. Bảng NhanVien:Lưu thông tin nhân viên

* *Bảng lưu trữ dữ liệu*

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã đại lí** | **Tên đại lý** | **SĐT** | **Địa chỉ** |
| *1* | *Honda* | *028343298* | *Bình Phước* |
| *2* | *Yamaha* | *023948348* | *Bình Thuận* |
| *3* | *Suzuki* | *023478789* | *Đồng Tháp* |

* *Bảng cấu trúc*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| *1* | *MaDL* | *Int* | *Primary key* | *Mã đại lý* |
| *2* | *TenDL* | *Nvarchar(50)* | *Not null* | *Tên đại lý* |
| *3* | *SDT* | *Char(10)* | *Not null* | *Số điện thoại* |
| *4* | *DiaChi* | *Nvarchar(50)* | *Not null* | *Địa chỉ* |

1. Bảng Sản phẩm: Lưu thông tin sản phẩm

* *Bảng lưu trữ dữ liệu*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mã SP** | **Tên SP** | **Mã loại** | **Ngày sản xuất** | **Thông số kỹ thuật** | **Số lô** | **Đơn giá** |
| *1* | *Honda SH mode* | *1* | *11/2/2022* | *150cc* | *1231243* | *80000000* |
| *2* | *Yamaha exciter* | *2* | *1/3/2022* | *153cc* | *23432* | *45000000* |
| *3* | *GSX150cc* | *3* | *3/2/2022* | *149cc* | *23423* | *70000000* |

* *Bảng cấu trúc*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| *1* | *MaSP* | *Int* | *Primary key* | *Mã sản phẩm* |
| *2* | *TenSP* | *Nvarchar(50)* | *Not null* | *Tên sản phẩm* |
| *3* | *MaLoai* | *Int* | *Not null* | *Mã loại* |
| *4* | *NgaySX* | *Date* | *Not null* | *Ngày sản xuất* |
| *5* | *ThongSoKT* | *Nvarchar(50)* | *Not null* | *Thông số kĩ thuật* |
| *6* | *SoLo* | *Int* |  | *Số lô SX* |
| *7* | *DonGia* | *Int* |  | *Giá bán* |

1. Bảng HoaDonBan: Lưu thông tin hóa đơn bán

* *Bảng lưu trữ dữ liệu*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mã HD** | **Ngày bản** | **Tên khách** | **SĐT** | **Địa chỉ** | **Mã NV** | **Tổng tiền** |
| *1* | *11/2/2022* | *Nguyễn Văn Xuân* | *028343298* | *Bình Phước* | *1* | *45000000* |
| *2* | *1/3/2022* | *Trần Văn Lan* | *023948348* | *Bình Thuận* | *2* | *60000000* |
| *3* | *3/2/2022* | *Hoàng Xuân Thạch* | *023478789* | *Đồng Tháp* | *3* | *70000000* |

* *Bảng cấu trúc*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| *1* | *MaHD* | *Int* | *Primary key* | *Mã hóa đơn* |
| *2* | *TenKhach* | *Nvarchar(50)* | *Not null* | *Tên khách* |
| *3* | *SDT* | *Char(10)* | *Not null* | *Số điện thoại* |
| *4* | *DiaChi* | *Nvarchar(50)* | *Not null* | *Địa chỉ* |
| *5* | *NgayBan* | *Date* |  | *Ngày bán* |
| *6* | *MaNV* | *Int* | *Not null* | *Nhân viên bán* |
| *7* | *TongTien* | *Int* |  | *Tổng thanh toán* |

1. Bảng NhanVien:Lưu thông tin nhân viên

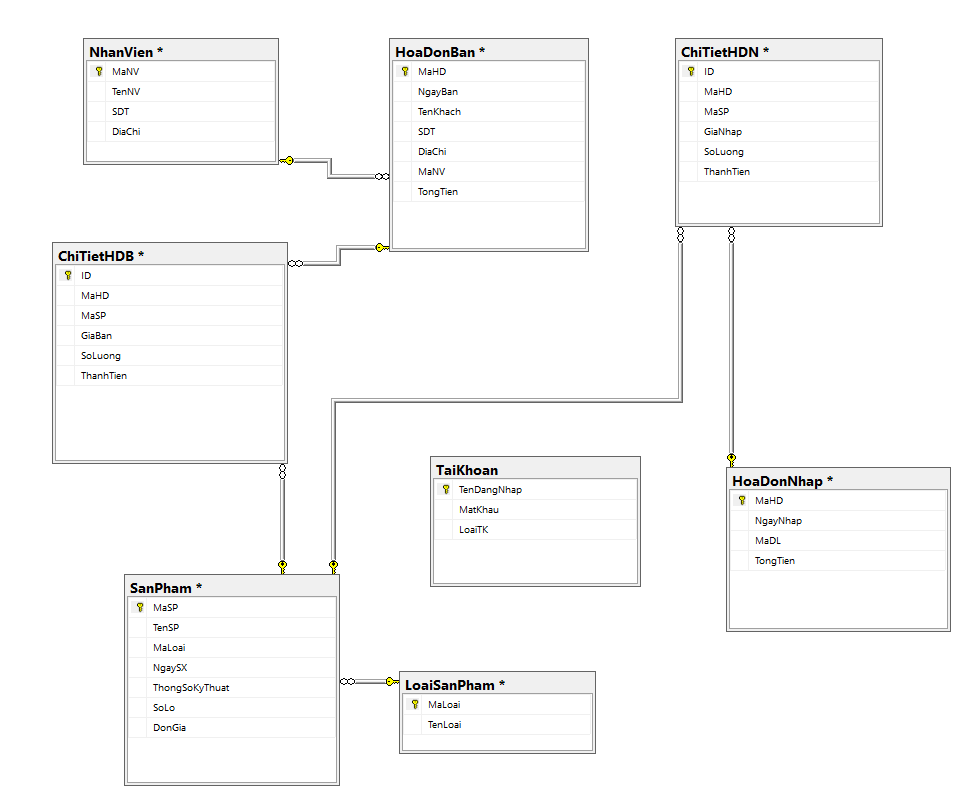
* *Bảng lưu trữ dữ liệu*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Mã hóa đơn** | **Mã sản phẩm** | **Số lượng** |
| *1* | *1* | *1* | *3* |
| *2* | *2* | *2* | *4* |
| *3* | *3* | *3* | *2* |

* *Bảng cấu trúc*

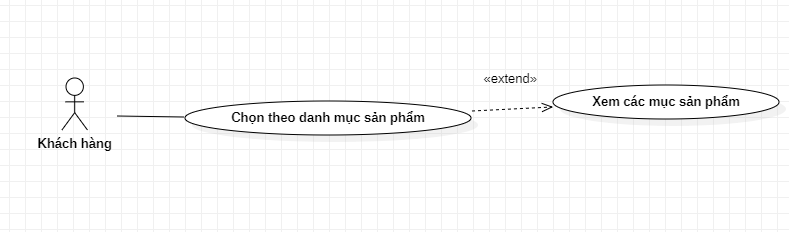
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| *1* | *ID* | *Int* | *Primary key* | *Mã CT hóa đơn* |
| *2* | *MaHD* | *Int* | *Not null* | *Mã hóa đơn* |
| *3* | *MaSP* | *Int* | *Not null* | *Mã sản phẩm* |
| *4* | *SoLuong* | *Int* | *Not null* | *Số lượng mua* |

* 1. Các ràng buộc toàn vẹn trên dữ liệu (nếu có)
* Ràng buộc tham chiếu
* MaLoai Bảng sản phẩm là khóa ngoại tham chiếu tới MaLoai Bảng Loại sản phẩm
* MaNV Bảng hóa đơn là khóa ngoại tham chiếu tới MaNV bảng Nhân viên
* MaSP Bảng Chi tiết hóa đơn là khóa ngoại tham chiếu tới MaSP bảng Sản phẩm
* MaHD Bảng chi tiết hóa đơn là khóa ngoại tham chiếu tới MaHD bảng Hóa đơn
  1. Diagram

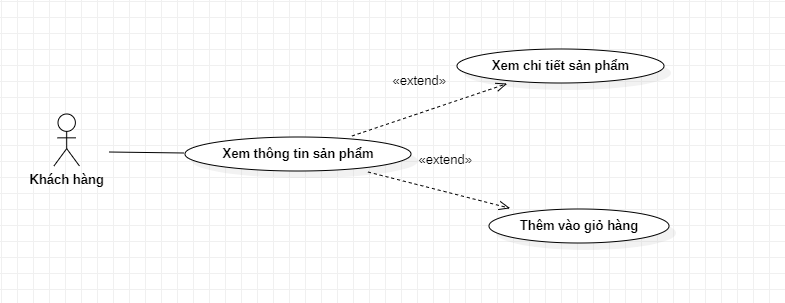


1. **Các giải thuật, module xử lý tiêu biểu**
   1. Sơ đồ phân rã chức năng hệ thống

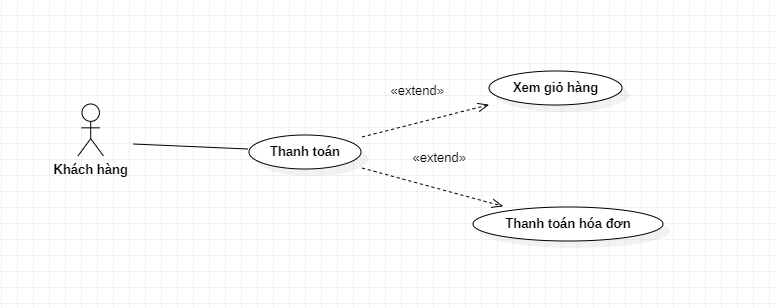
* Danh mục sản phẩm

**

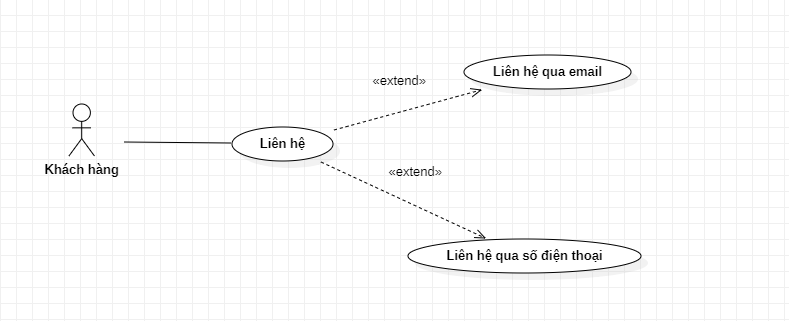
* Xem thông tin sản phẩm



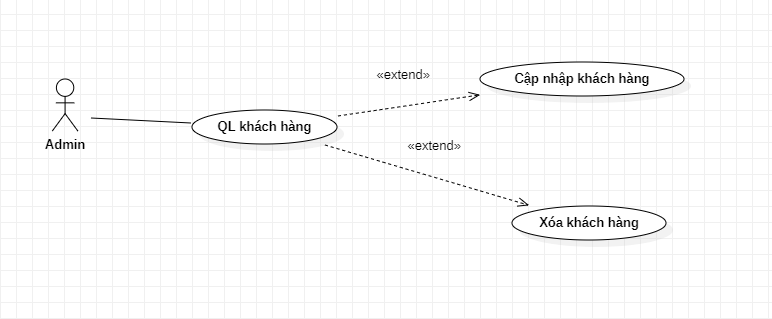
* Thanh toán



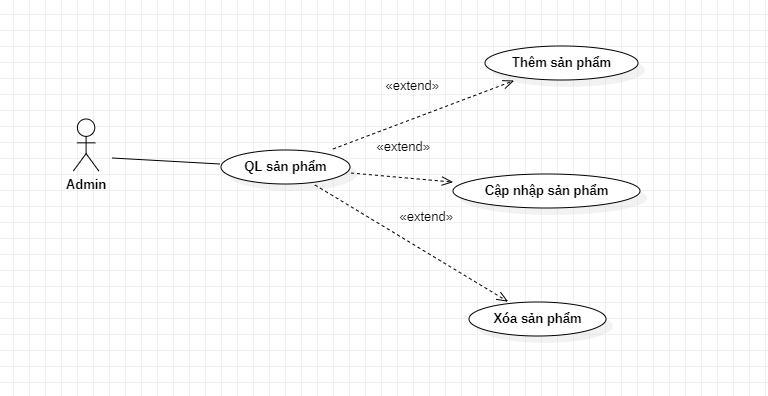
* Liên hệ

**

* Quản lí khách hàng

**

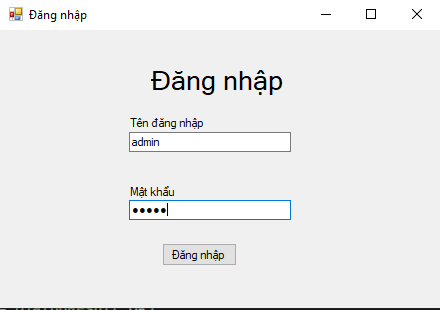
* Quản lí sản phẩm

**

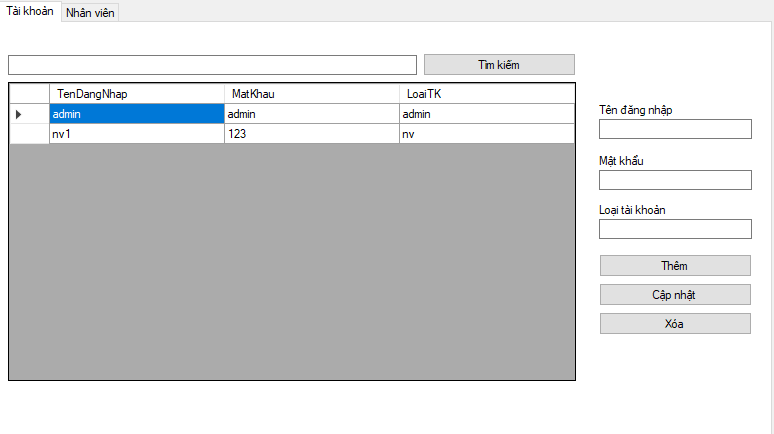
1. **Công cụ sử dụng xây dựng ứng dụng**

* Visual studio 2019
* SQL Server 2016

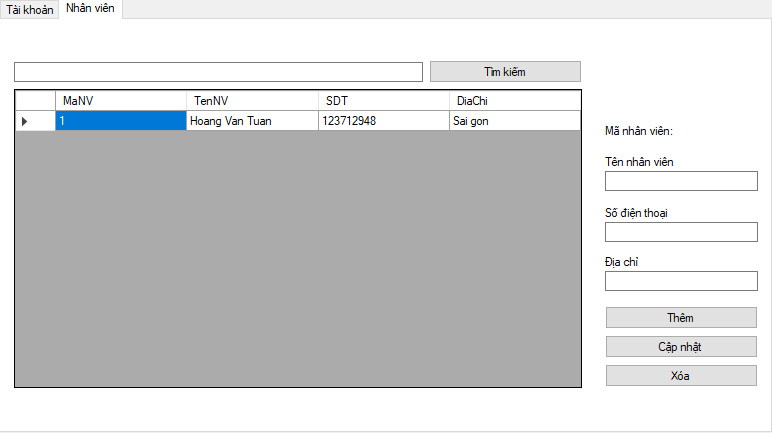
1. **Một số giao diện chính của chương trình**
2. Giao diện đăng nhập



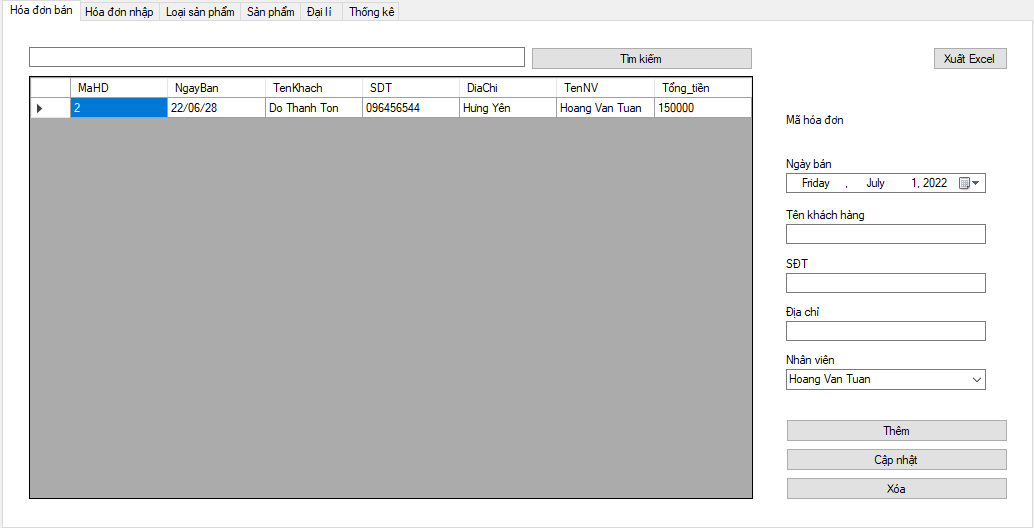
1. Giao diện quản lý tài khoản



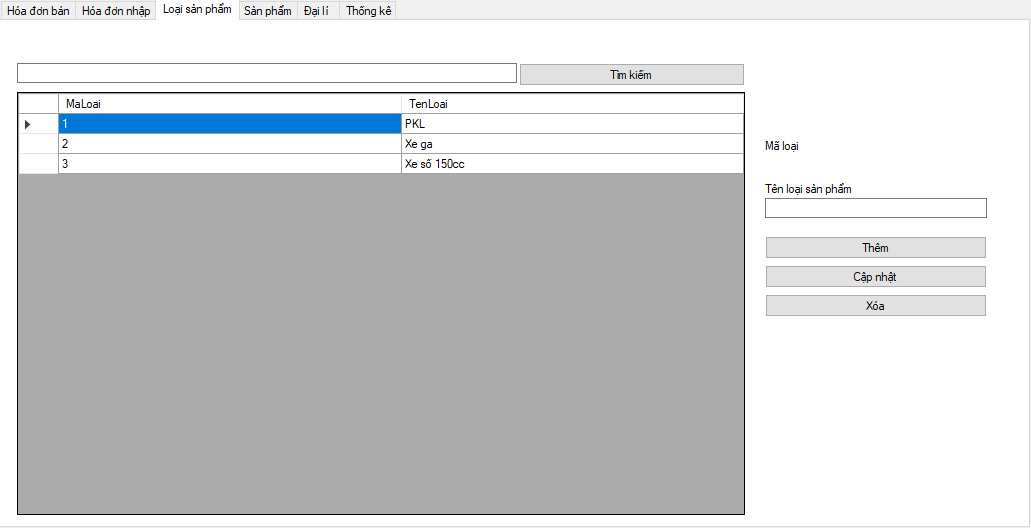
1. Giao diện quản lý nhân viên

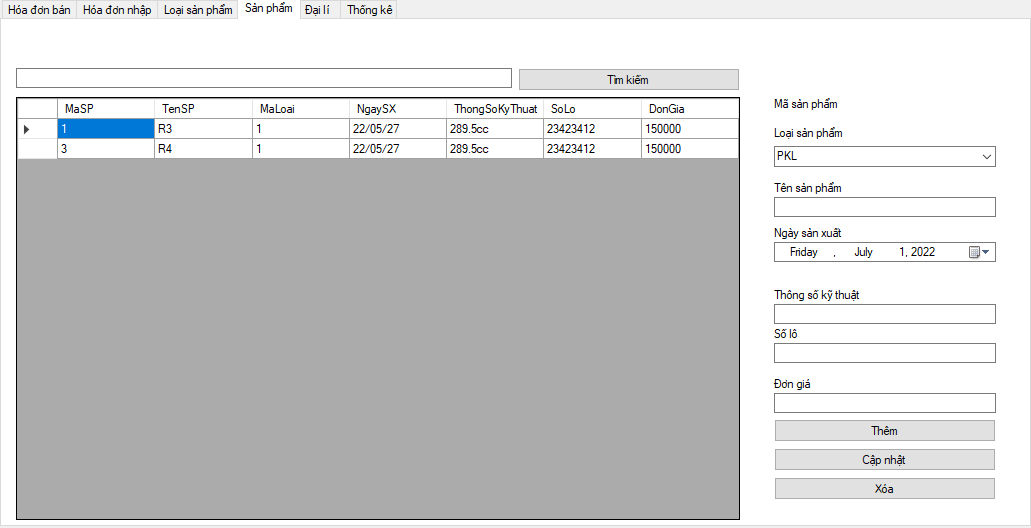


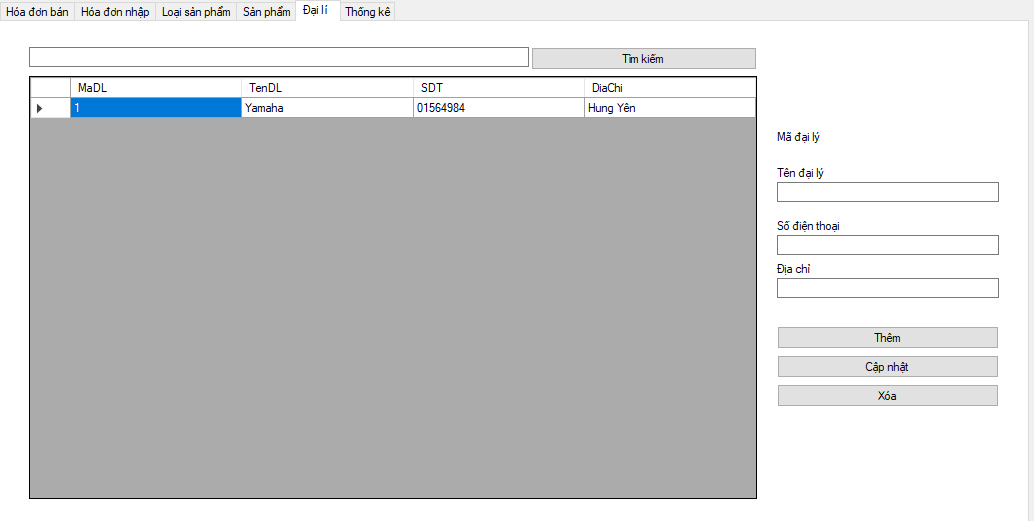
1. Các giao diện tiêu biểu khác













# BÀI TẬP CÁ NHÂN – SV1: Nguyễn Thị Hồng

**Bài tập chương 2 \_ P1: Lập trình Hướng đối tượng**

**1. Bài 3: Cài đặt class phân số thực hiện phép cộng, trừ nhân chia hai phân số**

- Trình bày giải thuật

Khởi tạo class PhanSo vơi từ số và mẫu số

Tìm ước chung lớn nhất của tử số và mẫu số

Rút gọn phân số

Hàm tính tổng: quy đồng mẫu số cộng tử số

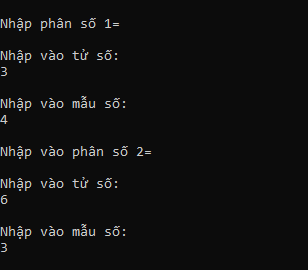
Hàm tính hiệu: quy đồng mẫu số và trừ số

Hàm tính tích: lấy từ nhân tử, mẫu nhân mẫu

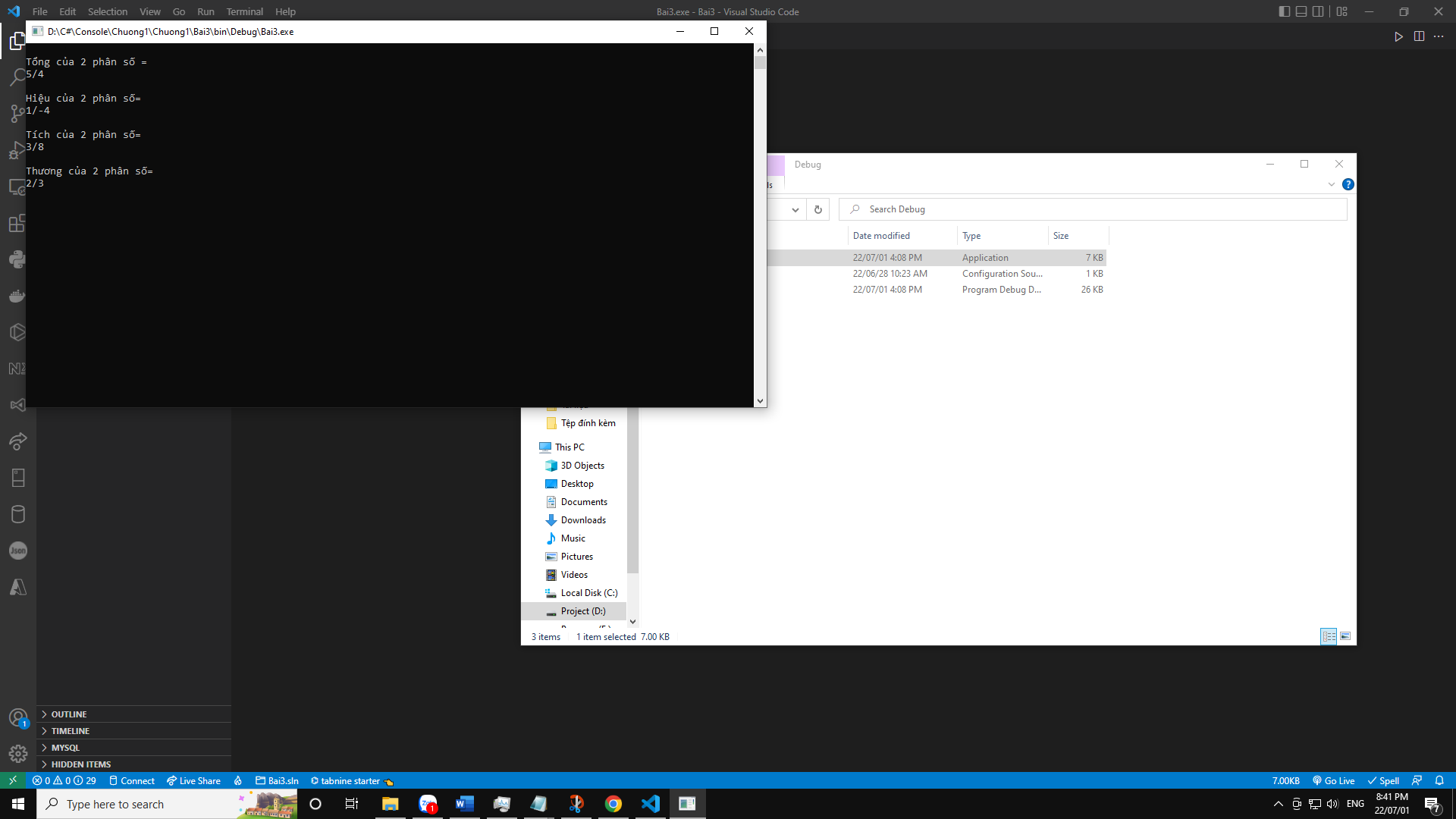
Hàm tính thương: lấy tử số thứ nhất nhân mẫu số thứ 2, mẫu số thứ nhất nhân tử số thứ 2

- Giao diện kết quả

Nhập phân số



Kết quả



**Bài tập chương 2\_P2: Tính kế thừa – đa hình trong HĐT**

**2. Bài 4. Viết chương trình tính lương Cán bộ của một trường học (sử dụng tính kế thừa).**

- Trình bày giải thuật

Khởi tạo đối tượng Cán bộ: Mã cb, họ tên, năm sinh, quê quán, lương cơ bản, hệ số lương

Phương thức nhập, hiện thông tin

Khởi tạo đối tượng CB văn phòng kế thừa từ đối tượng Cán bộ

Hàm tính lương: lương = (lương cơ bản\*hệ số lương)+phụ cấp(10% lương cơ bản)

Khởi tạo đối tượng CB Giảng viên kế thừa đối tượng Cán bộ

Tính phụ cấp giảng dạy, nếu số năm >5 phụ cấp = lương\*0.1, còn lại =0

Tính lương: lương = (lcb\*hsl)+phụ cấp(lcb\*0.4)+phụ cấp thâm niên

Cho nhập n, m số lượng cán bộ văn phòng và giảng viên

Khởi tạo mảng CanBo với độ dài : n+m

Nhập thông tin cho từng cán bộ

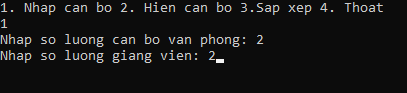
Duyệt danh sách cán bộ, gọi hàm hiện trong đối tượng

Sắp xếp:

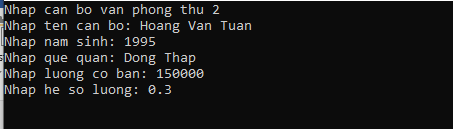
Sử dụng sắp xếp nối bọt, kiểm tra lương người trước có lớn hơn người sau không, nếu lớn hơn thì đổi chỗ, lặp đi lặp lại cho đến khi danh sách được duyệt hết

- Trình bày các sự kiện

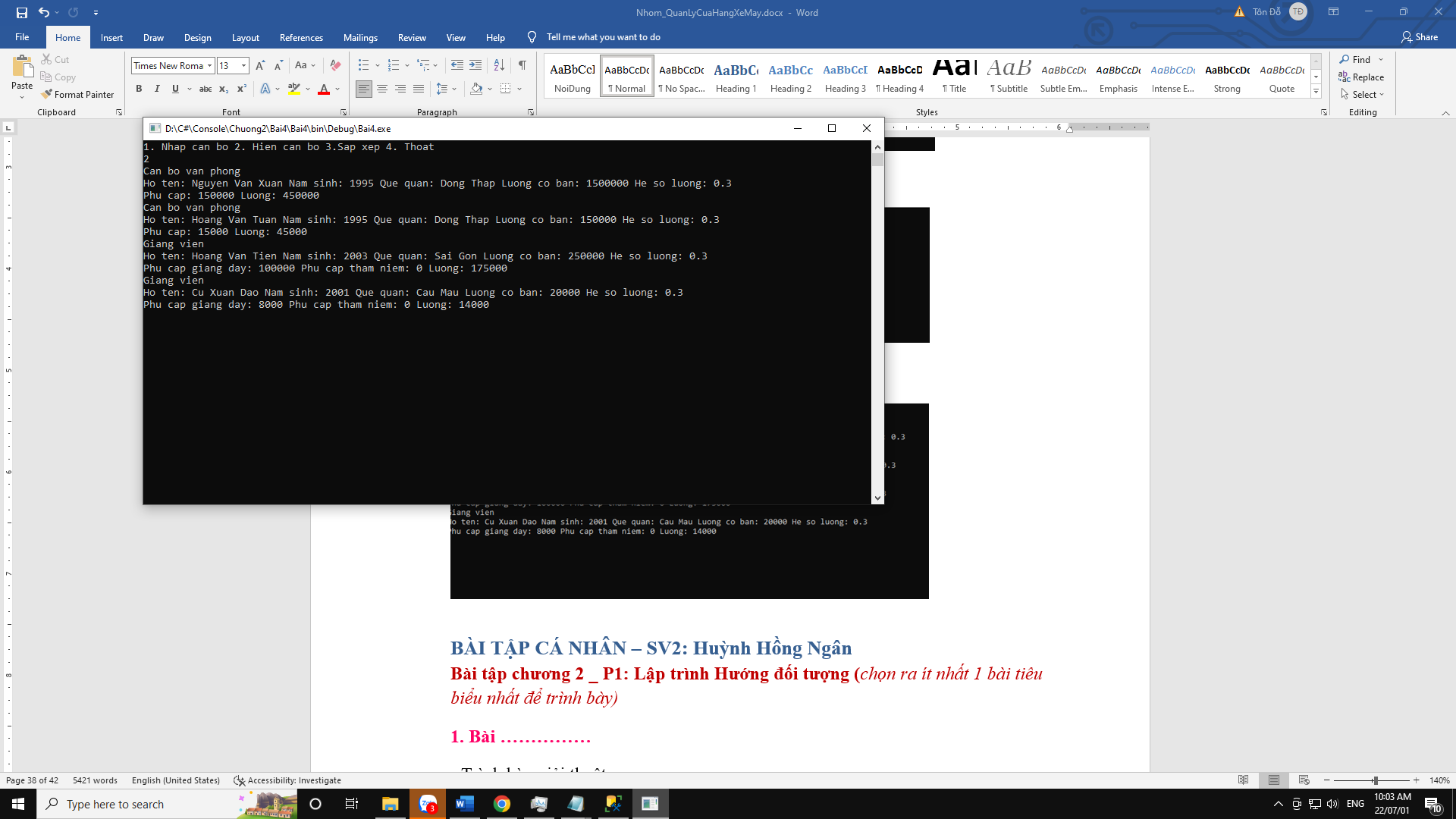
Nhập số lượng cán bộ:



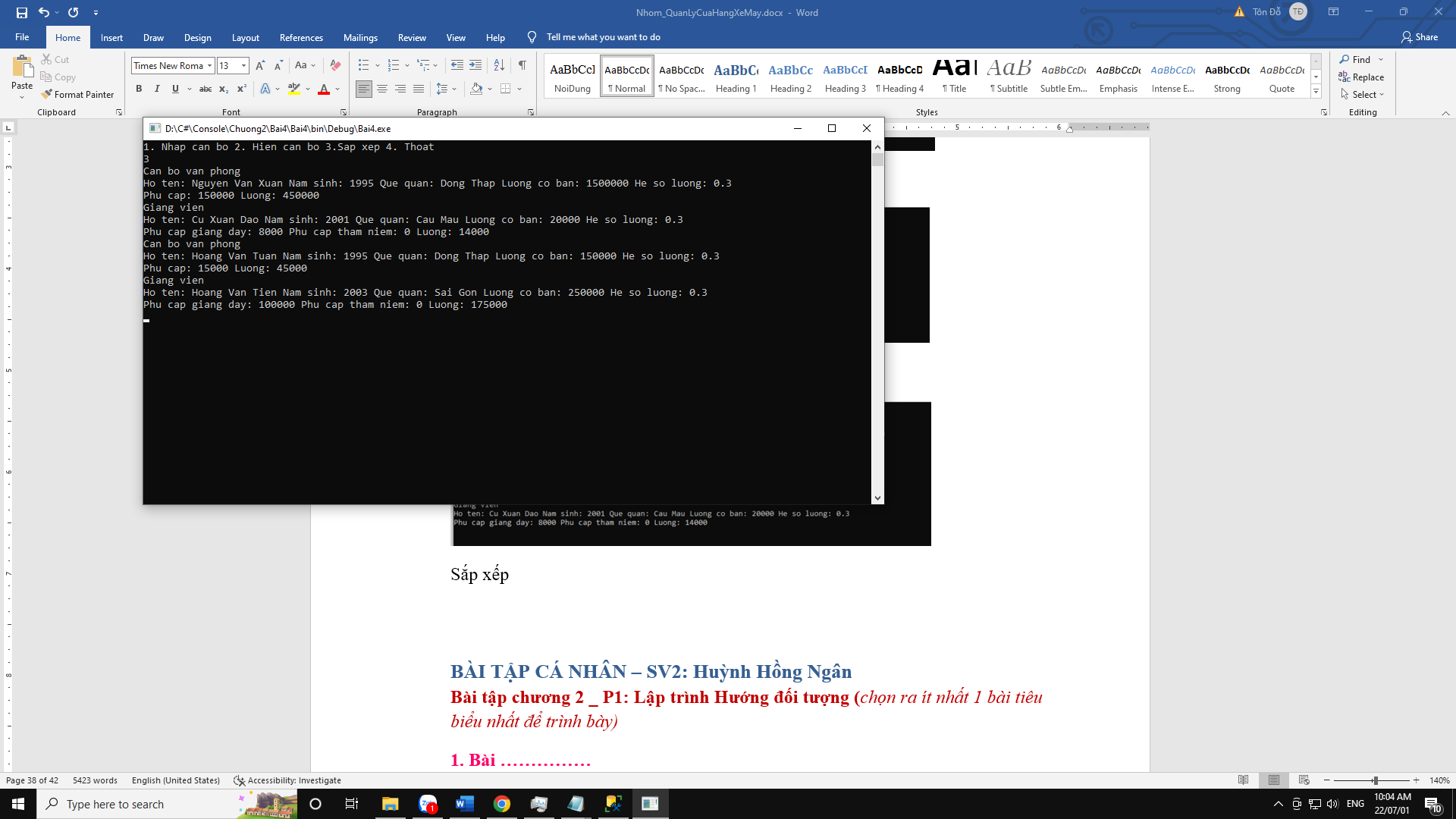
Nhập thông tin cán bộ



Hiện cán bộ



Sắp xếp



# BÀI TẬP CÁ NHÂN – SV2: Huỳnh Thị Hồng Ngân

**Bài tập chương 2 \_ P1: Lập trình Hướng đối tượng**

**1. Bài 4: Thiết kế class Môn học với các dữ liệu cần lưu là: Mã môn, tên môn, số tín chỉ**

- Trình bày giải thuật

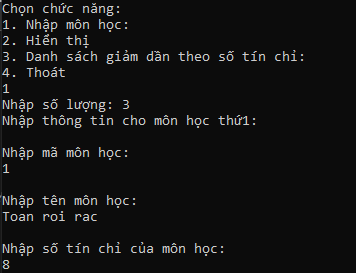
Khởi tạo class Môn học gồm mã môn, tên môn, số tc

Tính số lượng tiết học = số tc\* số tiết/tc

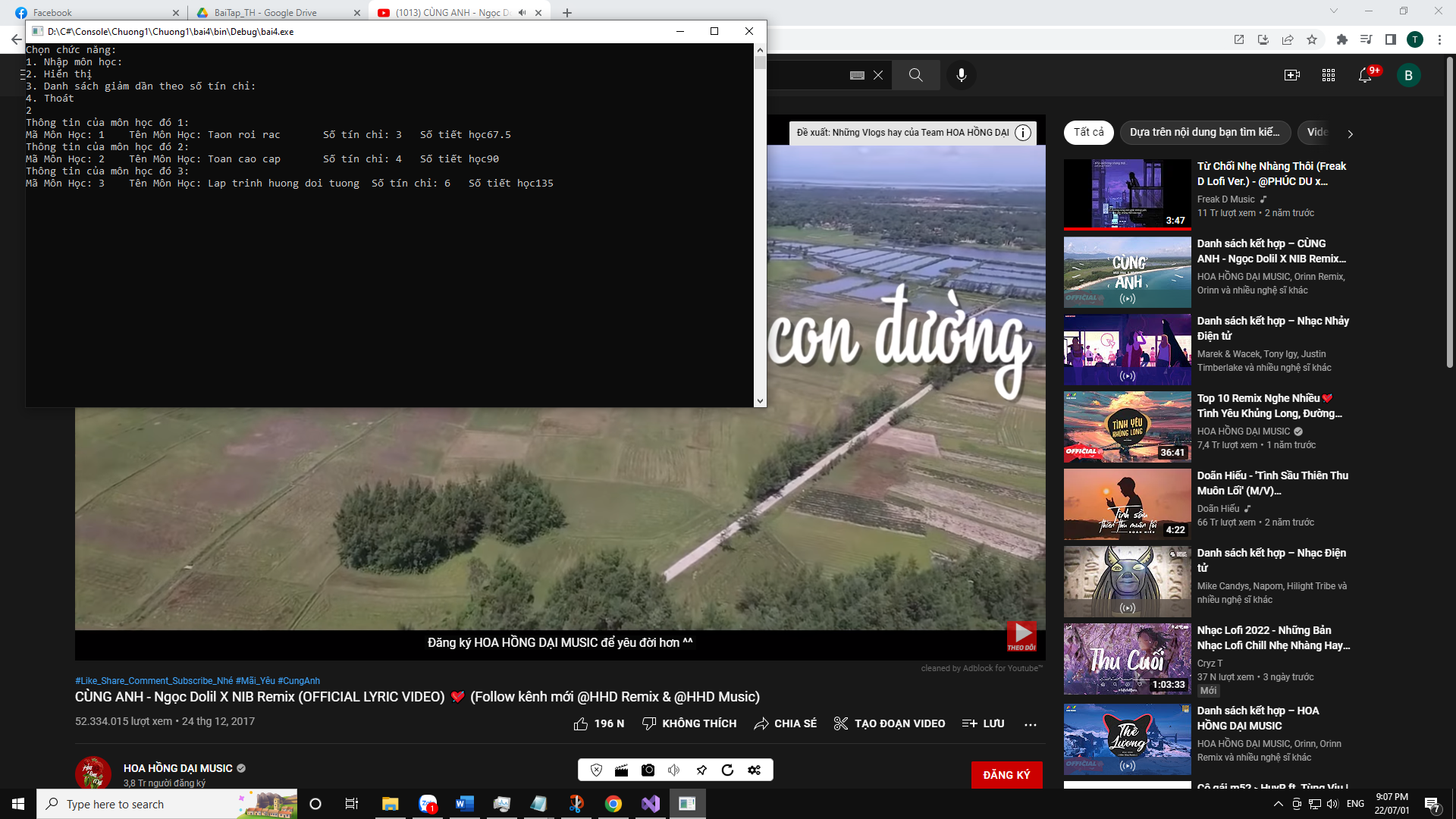
Sử dụng giải thuật sắp xếp nổi bọt, duyệt danh sách môn học

- Giao diện kết quả

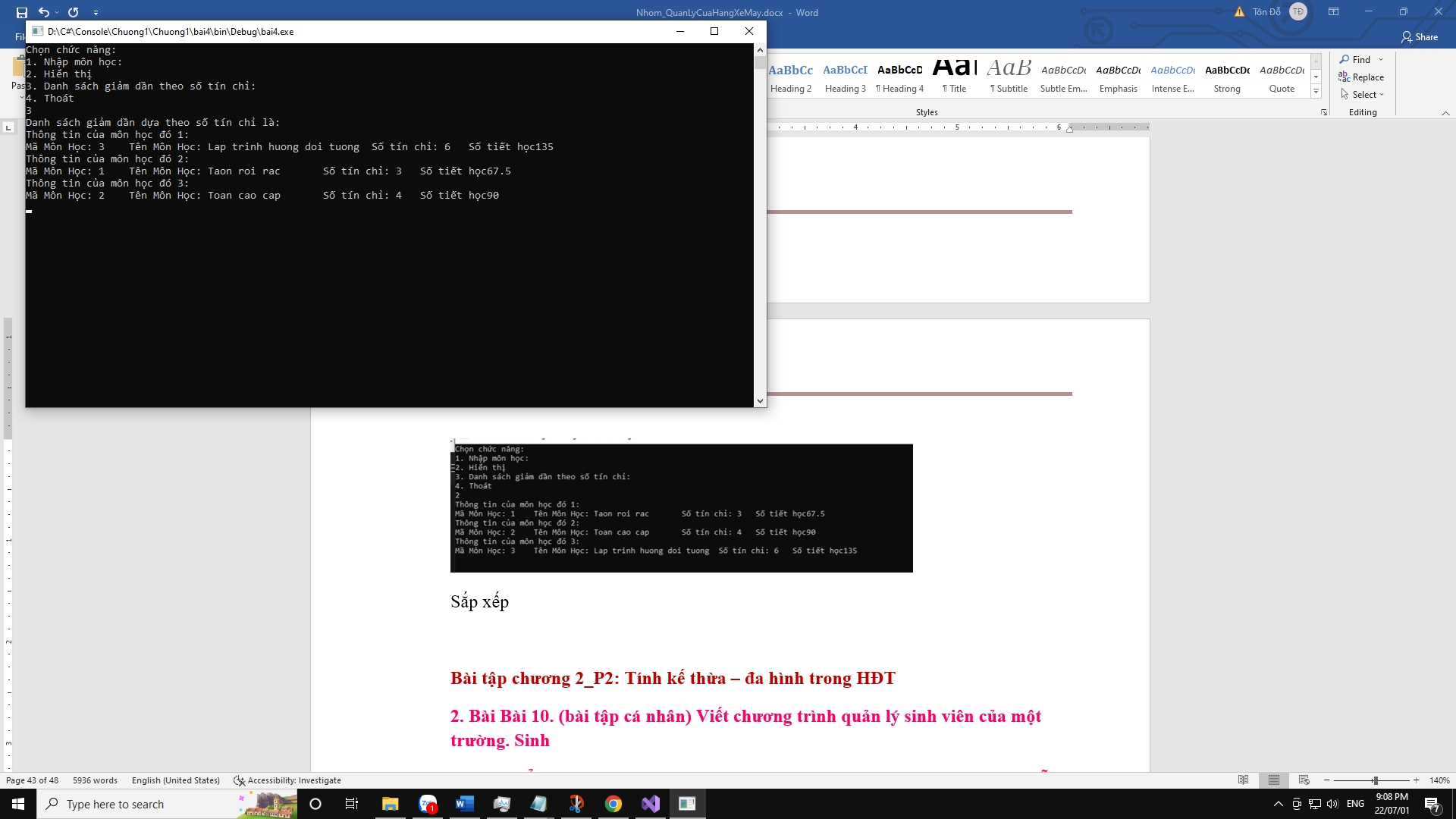
Nhập môn học



Hiển thị danh sách môn học



Sắp xếp



**Bài tập chương 2\_P2: Tính kế thừa – đa hình trong HĐT**

**2. Bài 3. Xây dựng chương trình quản lý độc giả của thư viện**

- Trình bày giải thuật

Khởi tạo đối tượng giảng viên: mã giảng viên, tên giảng viên, giơi tính, chuyên ngành

Khởi tạo đối tượng sinh viên: mã sinhv iên, tên sv, giới tính, lớp khoa

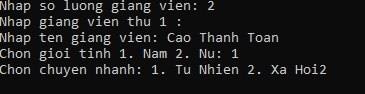
Viết hàm hiện và nhập thông tin từng đối tượng

Thống kê:

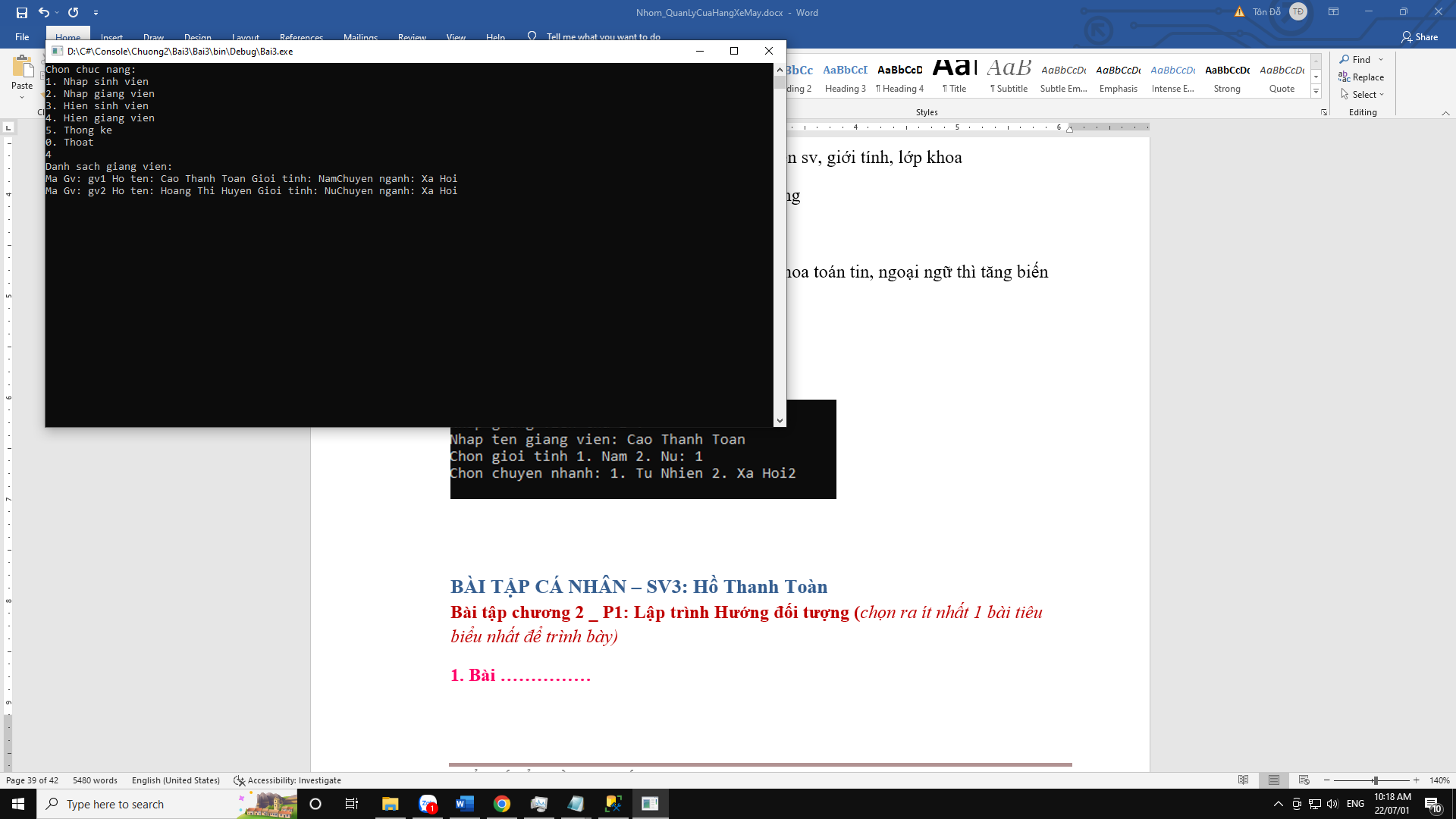
Duyệt danh sách sinh viên, nếu sinh viên là khoa toán tin, ngoại ngữ thì tăng biến đếm lên 1

- Trình bày các sự kiện

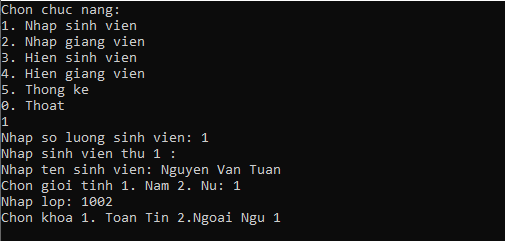
Nhập thông tin giảng viên



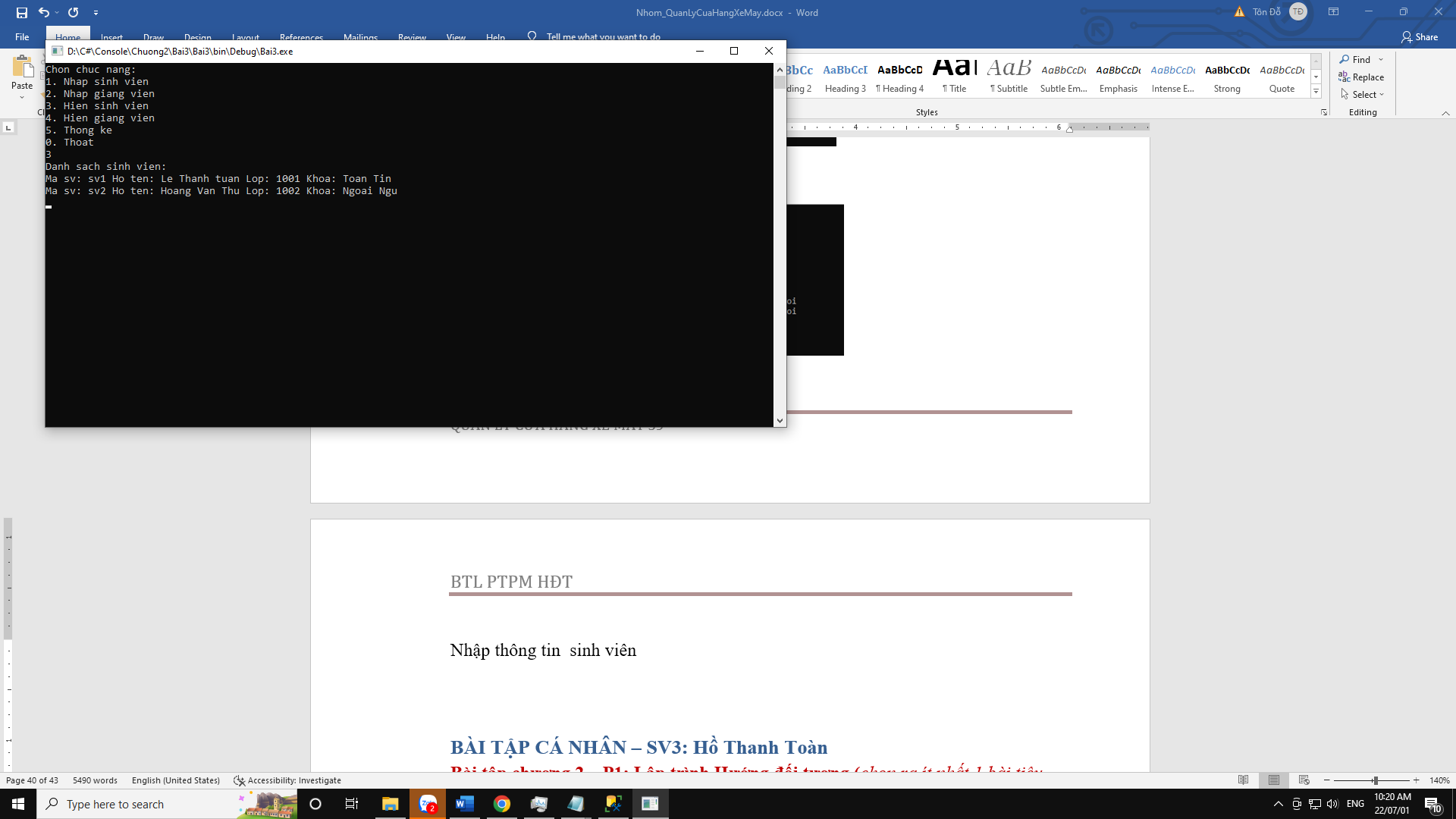
Hiện thông tin giảng viên



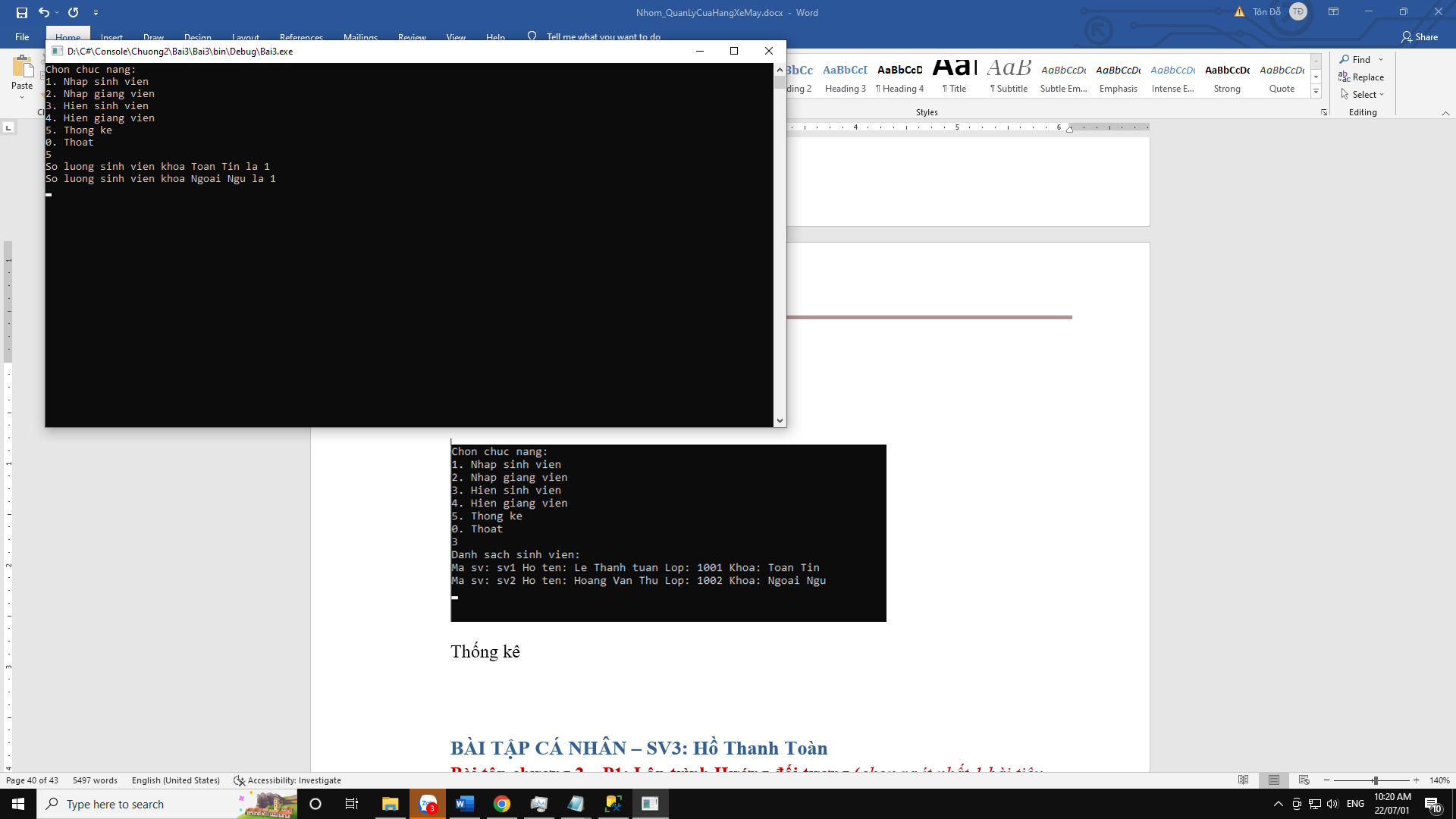
Nhập thông tin sinh viên



Hiện thông tin sinh viên



Thống kê



# BÀI TẬP CÁ NHÂN – SV3: Hồ Thanh Toàn

**Bài tập chương 2 \_ P1: Lập trình Hướng đối tượng**

**1. Bài 4: Thiết kế class Môn học với các dữ liệu cần lưu là: Mã môn, tên môn, số tín chỉ**

- Trình bày giải thuật

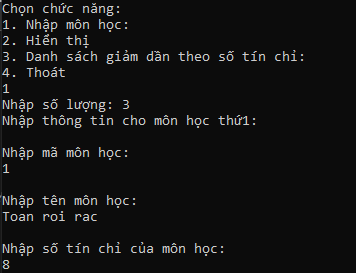
Khởi tạo class Môn học gồm mã môn, tên môn, số tc

Tính số lượng tiết học = số tc\* số tiết/tc

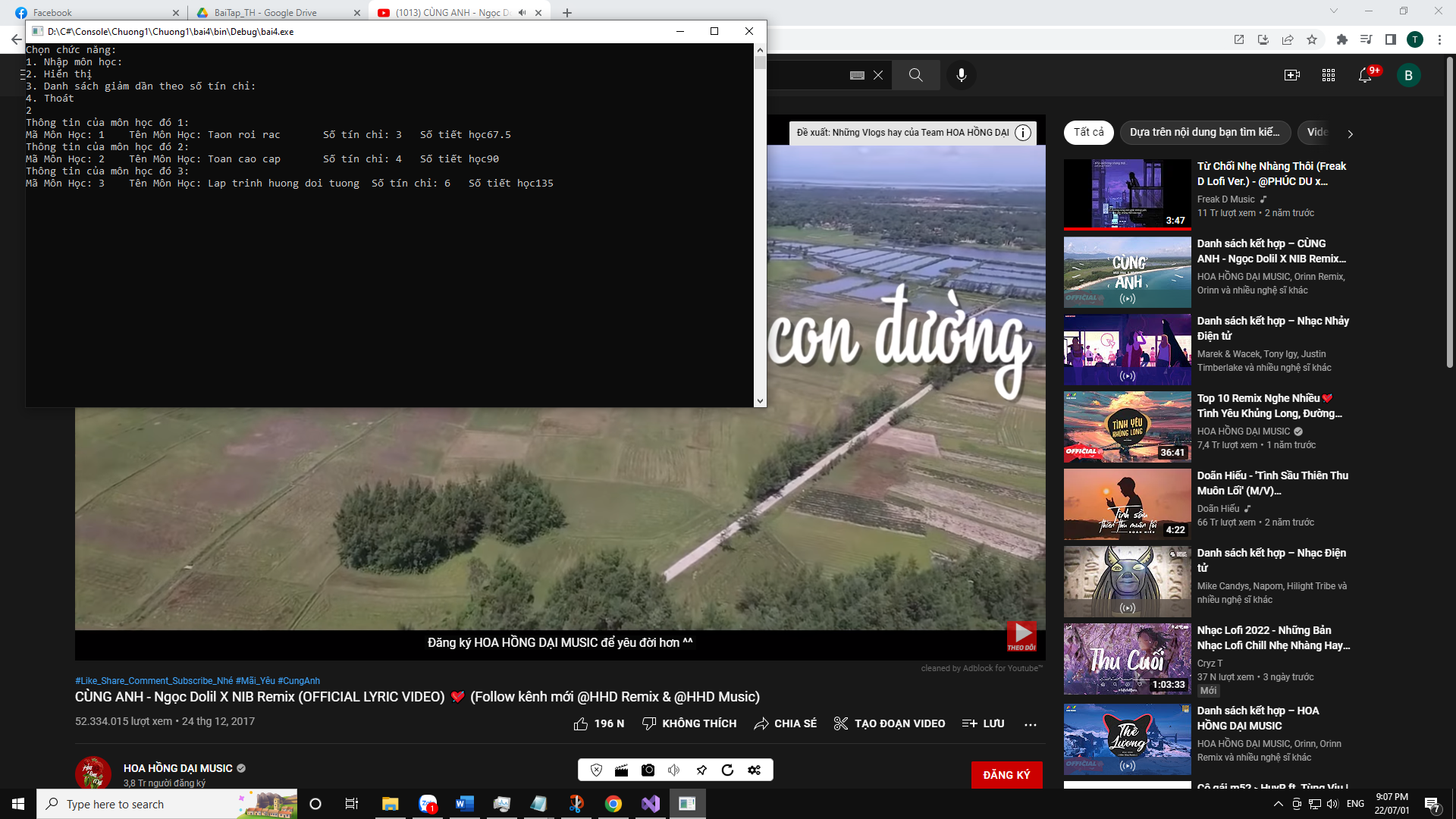
Sử dụng giải thuật sắp xếp nổi bọt, duyệt danh sách môn học

- Giao diện kết quả

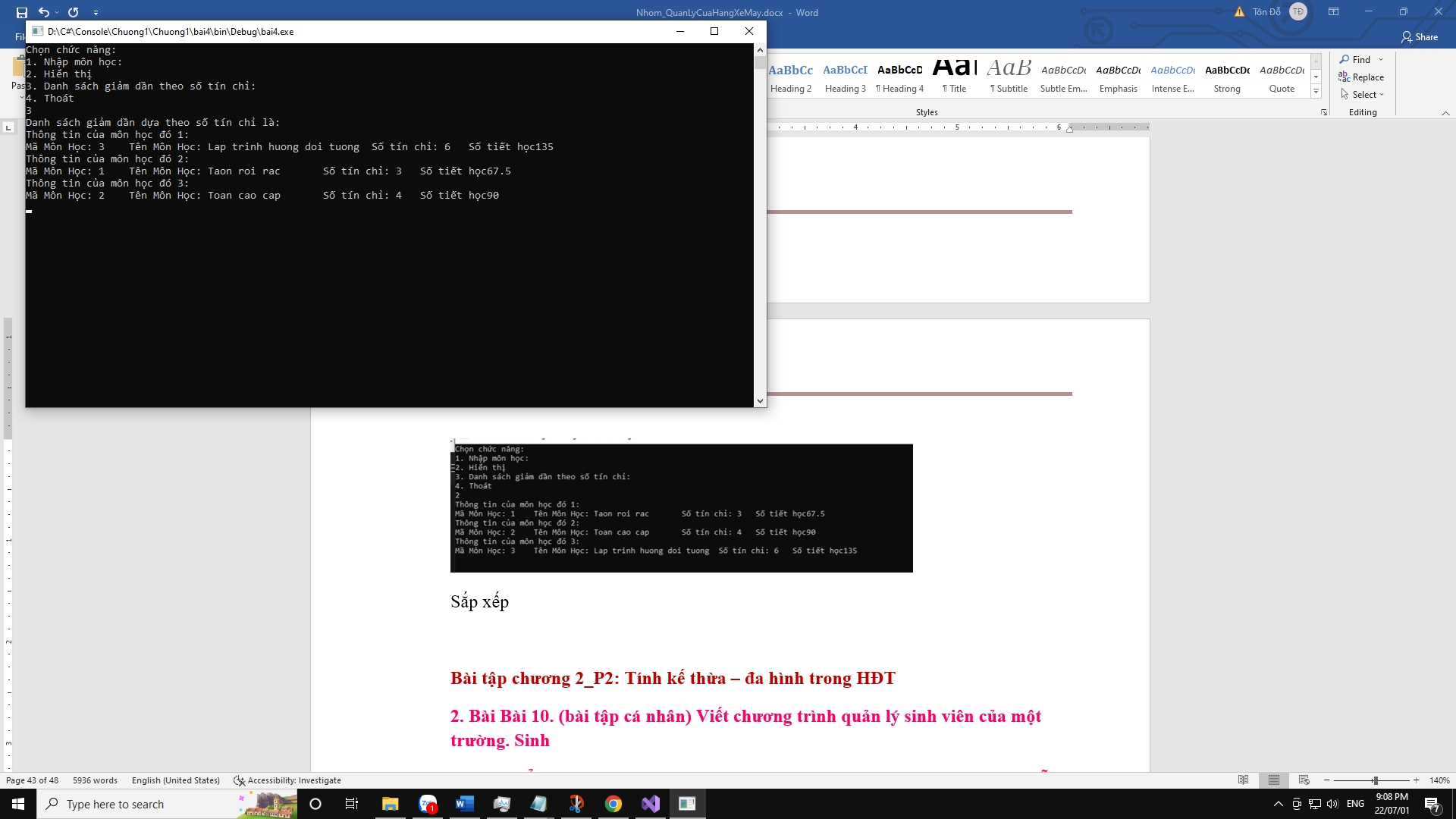
Nhập môn học



Hiển thị danh sách môn học



Sắp xếp



**Bài tập chương 2\_P2: Tính kế thừa – đa hình trong HĐT**

**2. Bài 10. (bài tập cá nhân) Viết chương trình quản lý sinh viên của một trường. Sinh**

**viên có thể học các chuyên ngành Công nghệ Thông tin, Vật lý, Ngữ văn. Mỗi**

**chuyên ngành tương ứng có các môn học khác nhau.**

- Trình bày giải thuật

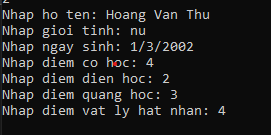
Khởi tạo đối tượng Sinh Viên gồm họ tên, giới tính, ngày sinh, có phương thức ảo Dtb

Khởi tạo các đối tượng SinhVienCNTT có thêm điểm csharp,pascal,sql. SinhVienVan có coDien,hienDai, SinhVienVL có: co,dien,hatNhan,quang

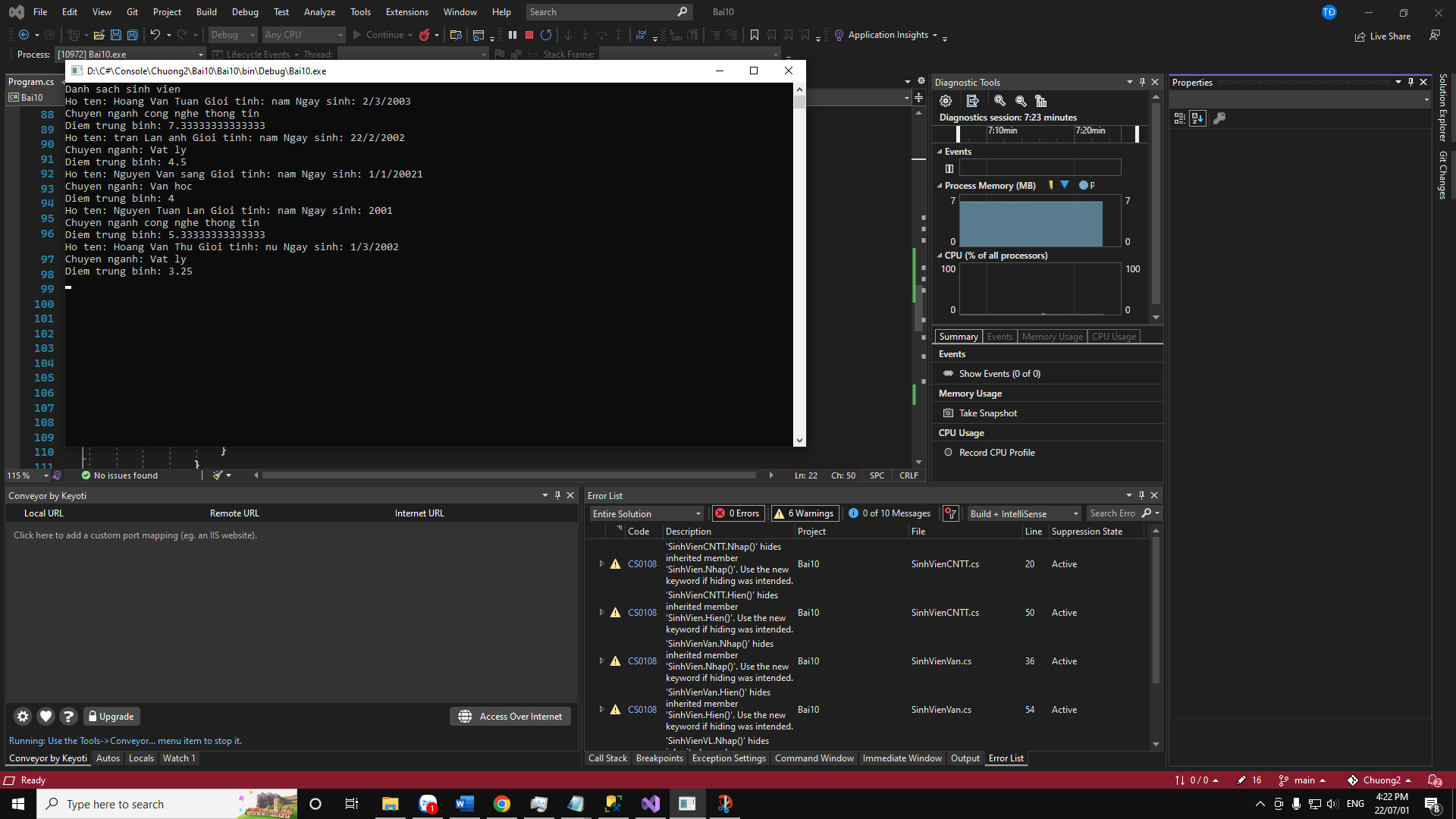
Hàm điểm trung bình bằng tổng điểm chia cho số môn

- Trình bày các sự kiện

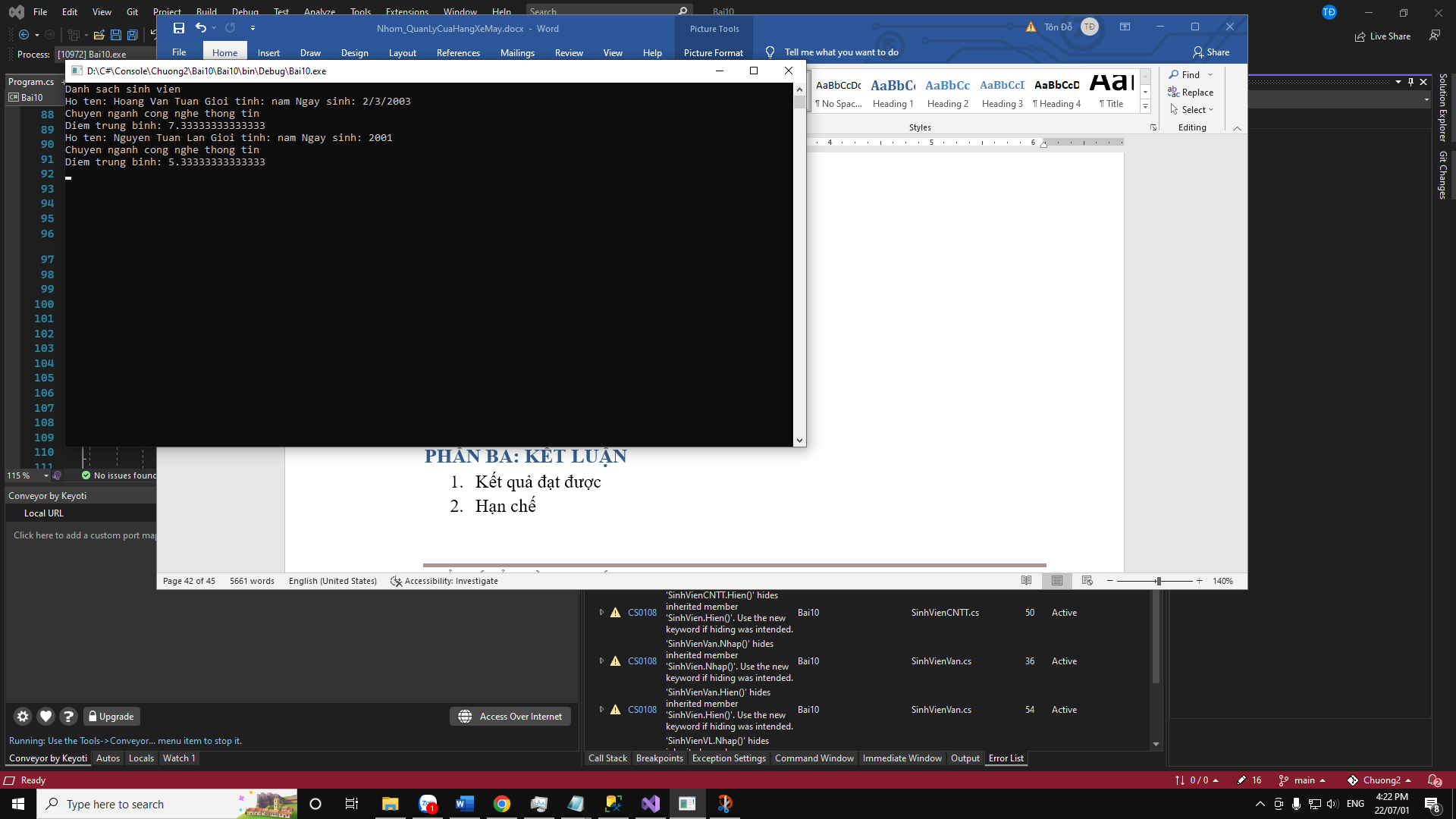
Nhập thông tin sinh viên



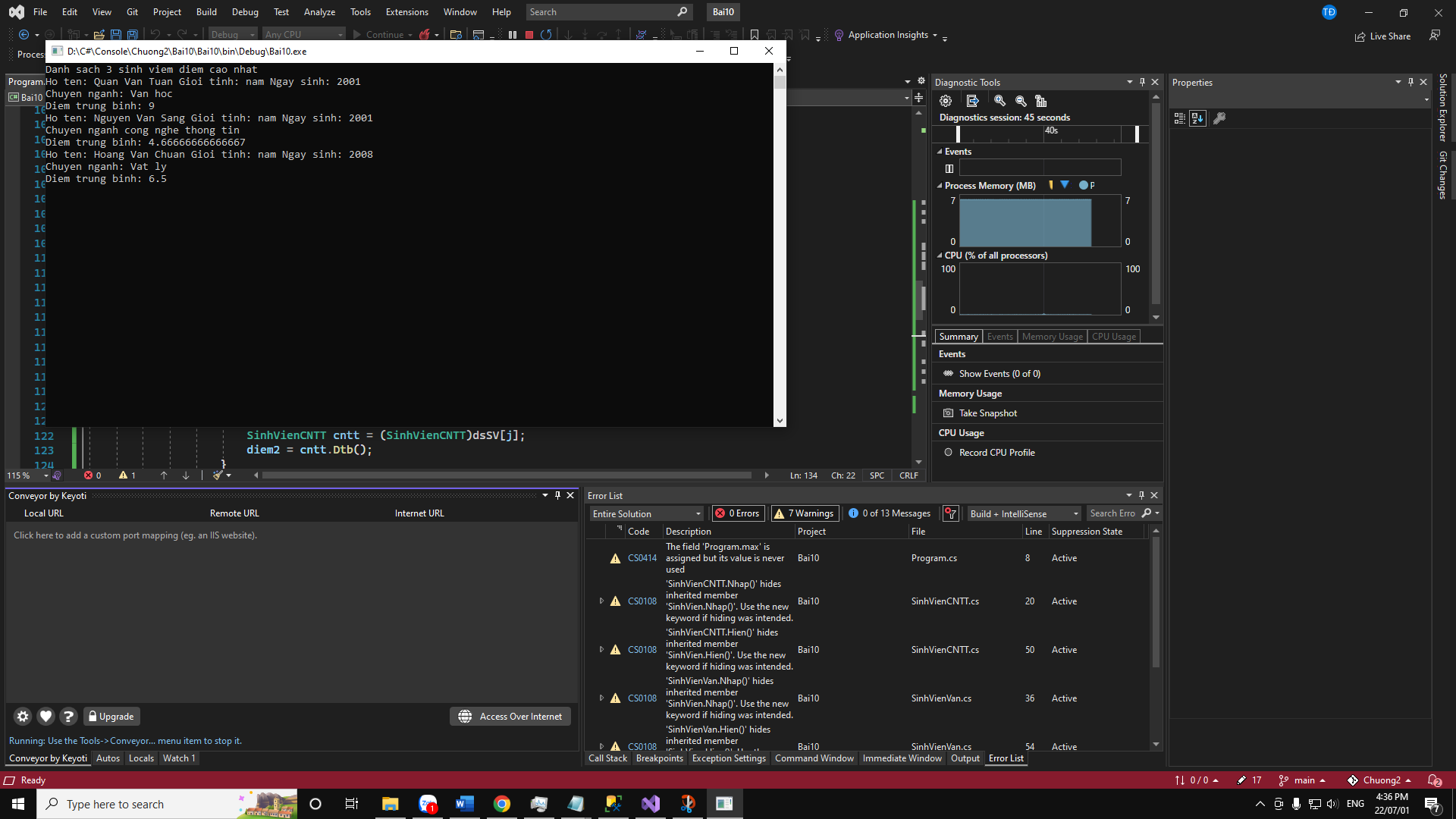
Hiển thị danh sách sinh viên



Hiện danh sách sinh viên điểm trung bình >5



Danh sách 3 sinh viên điểm cao nhất



# BÀI TẬP CÁ NHÂN – SV4: Phan Thị Cẩm Tú

**Bài tập chương 2 \_ P1: Lập trình Hướng đối tượng**

**1. Bài 3: Cài đặt class phân số thực hiện phép cộng, trừ nhân chia hai phân số**

- Trình bày giải thuật

Khởi tạo class PhanSo vơi từ số và mẫu số

Tìm ước chung lớn nhất của tử số và mẫu số

Rút gọn phân số

Hàm tính tổng: quy đồng mẫu số cộng tử số

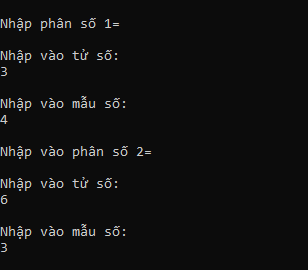
Hàm tính hiệu: quy đồng mẫu số và trừ số

Hàm tính tích: lấy từ nhân tử, mẫu nhân mẫu

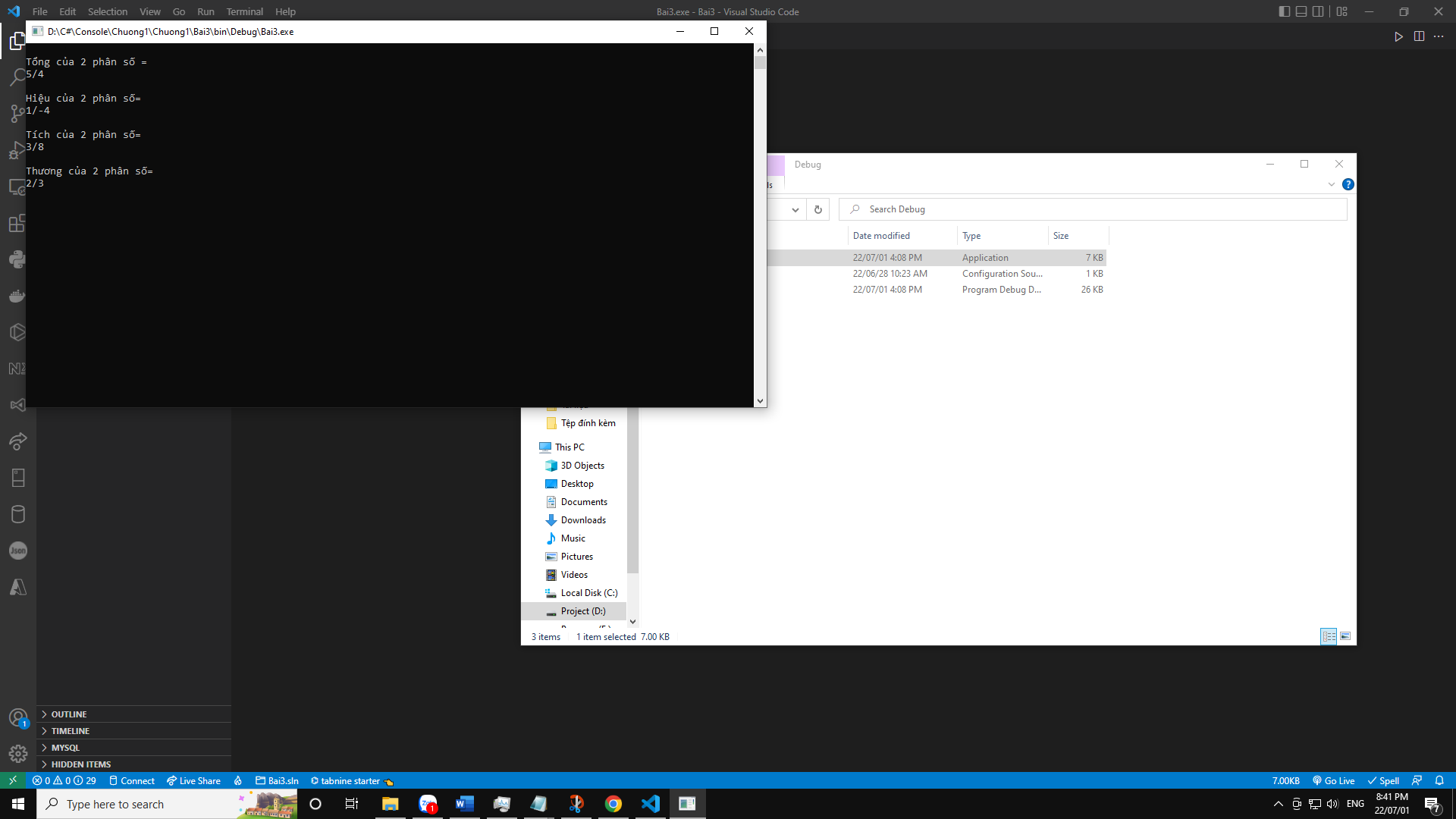
Hàm tính thương: lấy tử số thứ nhất nhân mẫu số thứ 2, mẫu số thứ nhất nhân tử số thứ 2

- Giao diện kết quả

Nhập phân số



Kết quả



**Bài tập chương 2\_P2: Tính kế thừa – đa hình trong HĐT**

**Bài 8. Viết chương trình tính diện tích và chu vi của Đa giác. Đa**

**giác được phân thành nhiều loại: Tam giác, hình chữ nhật, hình vuông.**

- Trình bày giải thuật

Khởi tạo class abtract DaGiac với so canh và mảng chiều dài cạnh

Viết hàm nhập chiều dài cạnh, kiểm tra cạnh phải lớn hơn 0

Khởi tạo class TamGiac kế thừa DaGiac

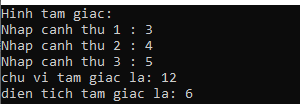
Viết hàm tính chu vi và diện tích theo công thức

Khởi tạo class HinhChuNhat kế thừa DaGiac

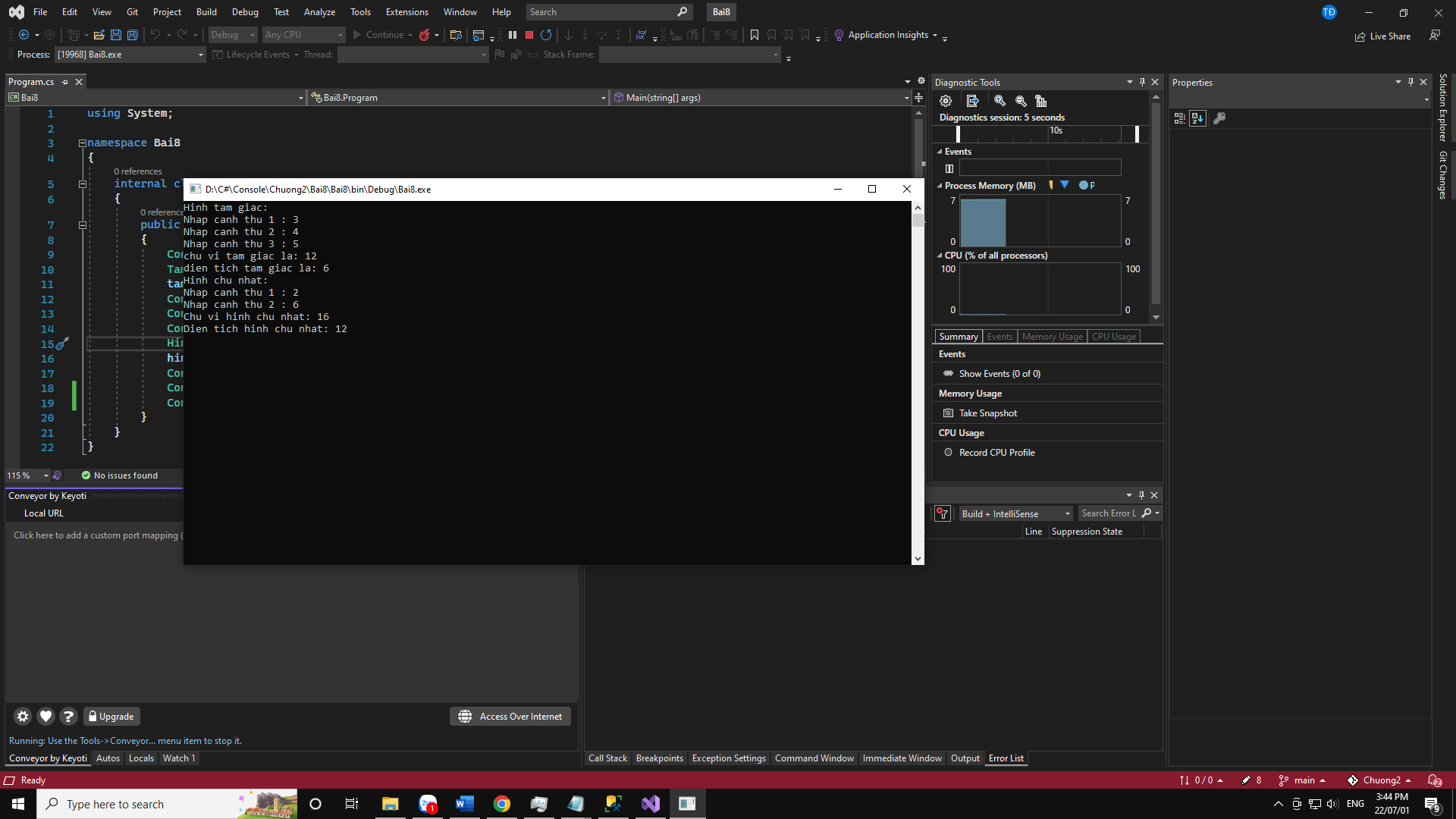
Viết hàm tính chu vi và diện tích

- Trình bày các sự kiện

Tính diện tích và chu vi hình tam giác



Tính diện tích và chu vi hình chữ nhật



# PHẦN BA: KẾT LUẬN

1. Kết quả đạt được

* Đã xây dựng thành công ứng dụng Quản lý cửa hàng xe máy winform đầy đủ yêu cầu cơ bản
* Nắm vững lập trình hướng đối tượng
* Nắm vững lập trình winform

1. Hạn chế

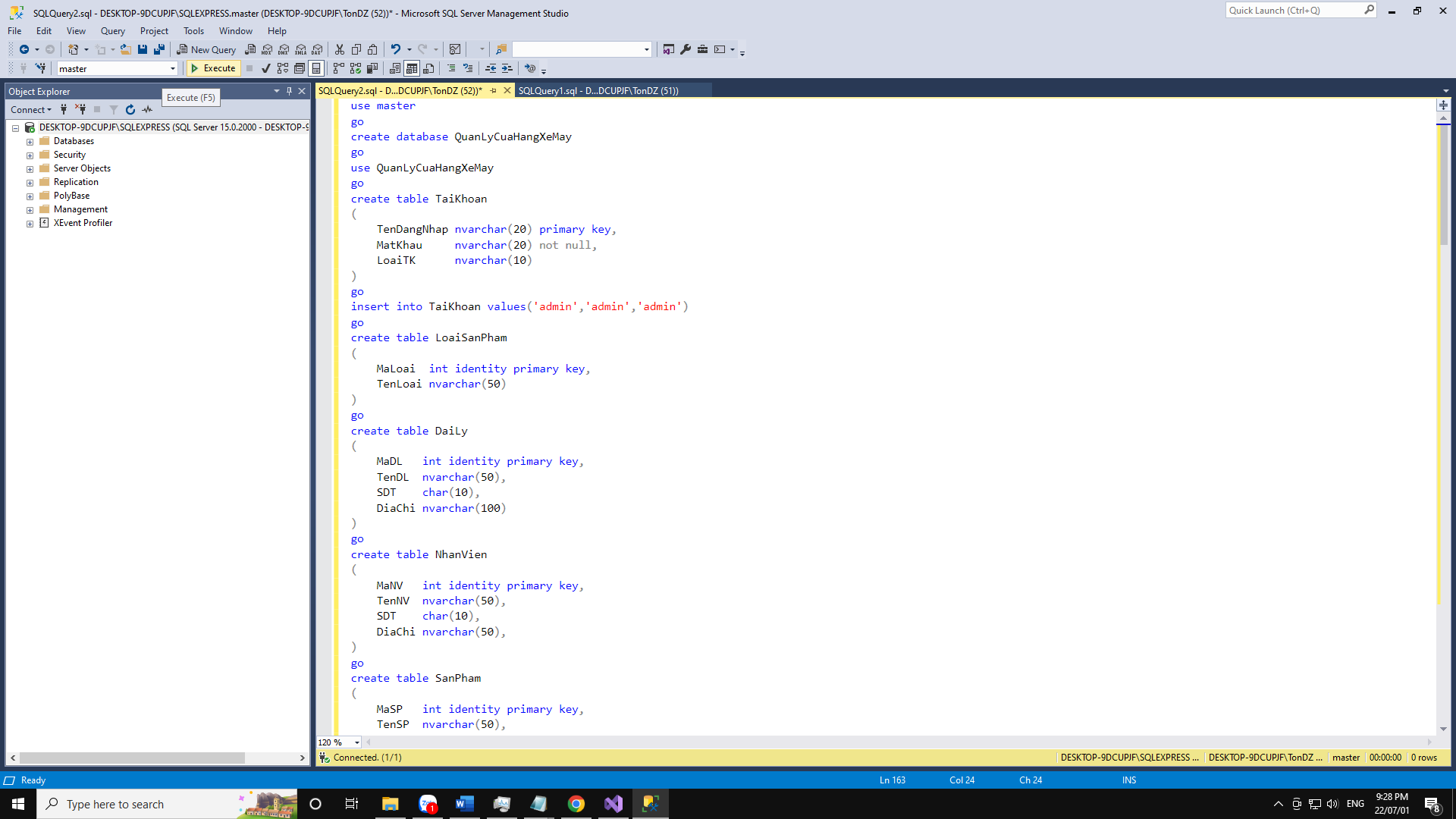
* Ứng dụng còn nhiều thiếu xót, đôi khi bị crash
* Giao diện chưa được đẹp mắt

1. Hướng phát triển

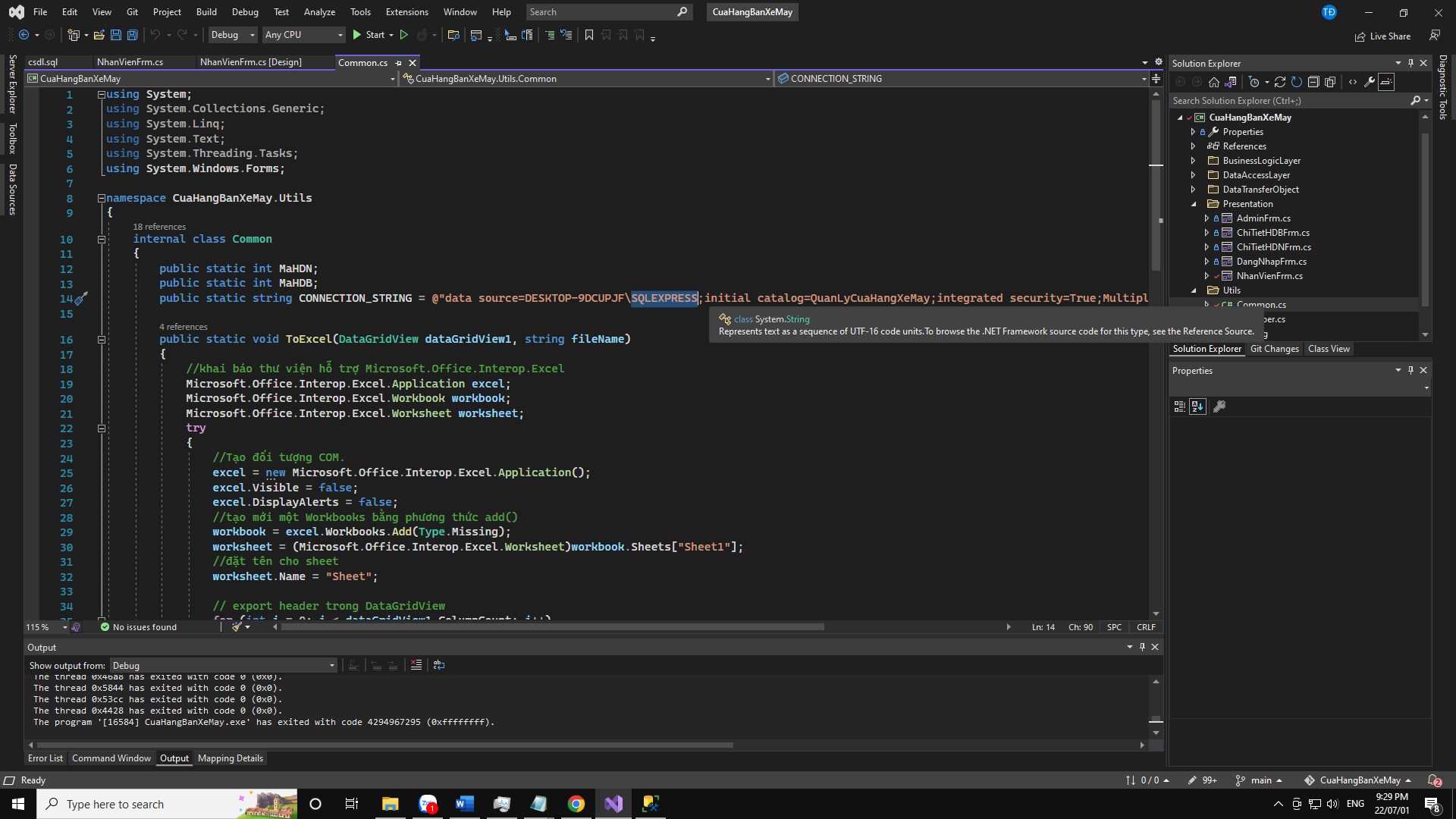
* Khắc phục một vài lỗi còn tồn đọng
* Thiết kế giao diện bắt mắt hơn

# HƯỚNG DẪN SỬ DỤNG

* Mở file database.sql chạy bằng SSMS để tạo CSDL



* Mở code bằng Visual Studio
* Tìm tới thư mục Utils/Common
* Sửa server của SQL theo máy được cài đặt tại biết CONNECTION\_STRING



* Tiến hành chạy chương trình, đăng nhập với tài khoản admin, mật khẩu admin để thêm tài khoản đăng nhập cho nhân viên

# BẢNG PHÂN CÔNG CÔNG VIỆC (Bài tập nhóm)

|  |  |
| --- | --- |
| **Họ tên học viên** | **Nội dung công việc thực hiện** |
| 5420440160– Nguyễn Thị Hồng | Thiết kế CSDL |
| 5420440165-Huỳnh Thị Hồng Ngân | Viết câu lệnh truy vấn SQL |
| 5420440175-Hồ Thanh Toàn | Thiết kế giao diện winform |
| 5420440178-Phan Thị Cẩm Tú | Xử lý sự kiện winform |

# TÀI LIỆU THAM KHẢO

1. <https://howkteam.vn/course/lap-trinh-winform-co-ban-27>
2. <https://topdev.vn/blog/oop-la-gi/>
3. <https://xuanthulab.net/linq-trong-lap-trinh-c-net-thuc-hinh-vi-du-linq.html>
4. <https://stackoverflow.com/>