|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC HOA SEN**  **KHOA KHOA HỌC & CÔNG NGHỆ**  **BÁO CÁO ĐỀ TÀI CÔNG NGHỆ DỊCH VỤ WEB**  **Đề tài: Xây dựng service quản lý siêu thị**  **GVDH:Võ Hoàng Hải**  **HSTH: 1.** Nguyễn Hoàng Duy  **2.** Lê Quang Hà  **3.** Tạ Ngọc Đại  **Lớp:** QL081  **Khoá:** 2008  ***Tp.HCM, năm 20112007*** |

**NHẬN XÉT CỦA GIẢNG VIÊN**

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc311093991)

[1. Tìm hiểu lý thuyết: 2](#_Toc311093992)

[1.1. Tổng quan về WCF: 2](#_Toc311093993)

[1.2. Tại sao lại sử dụng WCF: 2](#_Toc311093994)

[1.3. Kiến trúc của WCF: 3](#_Toc311093995)

[1.3.1. Contracts: 4](#_Toc311093996)

[1.3.1.1. Service Contract: 4](#_Toc311093997)

[1.3.1.2. Data Contract: 9](#_Toc311093998)

[1.3.1.3. Message Contracts: 11](#_Toc311093999)

[1.3.2. Runtime service: 12](#_Toc311094000)

[1.3.3. Message: 13](#_Toc311094001)

[1.3.4. Host and activation: 13](#_Toc311094002)

[1.4. Mô hình lập trình WCF: 14](#_Toc311094003)

[1.4.1. Sử dụng phương pháp hướng đối tượng hay hướng dịch vụ? 14](#_Toc311094004)

[1.4.2. Service Model (Mô hình dịch vụ) 17](#_Toc311094005)

[1.4.3. Các phương pháp lập trình với WCF: 19](#_Toc311094006)

[1.4.3.1. Declarative programming (Phương pháp khai báo) 20](#_Toc311094007)

[1.4.3.2. Explicit programming (Phương pháp lập trình trực tiếp) 21](#_Toc311094008)

[1.4.3.3. Phương pháp sử dụng tập tin cấu hình 21](#_Toc311094009)

[2. Thiết kế chương trình: 23](#_Toc311094010)

[2.1. Mô tả chung: 23](#_Toc311094011)

[2.1.1. Tổng quan về chương trình: 23](#_Toc311094012)

[2.1.2. Chức năng và đối tượng sử dụng chương trình: 23](#_Toc311094013)

[2.2. Mô tả chi tiết: 24](#_Toc311094014)

[2.2.1. PersonService: 24](#_Toc311094015)

[2.2.1.1. ThemNV 24](#_Toc311094016)

[2.2.1.2. XoaNV 24](#_Toc311094017)

[2.2.1.3. SuaThongTinNV 24](#_Toc311094018)

[2.2.1.4. XemThongTinNV 25](#_Toc311094019)

[2.2.1.5. ThemTV 25](#_Toc311094020)

[2.2.1.6. XoaTV 25](#_Toc311094021)

[2.2.1.7. SuaThongTinTV 25](#_Toc311094022)

[2.2.1.8. XemThongTinTV 25](#_Toc311094023)

[2.2.2. ProductService: 26](#_Toc311094024)

[2.2.2.1. ThemHH 26](#_Toc311094025)

[2.2.2.2. XoaHH 26](#_Toc311094026)

[2.2.2.3. SuaThongTinHH 26](#_Toc311094027)

[2.2.2.4. XemThongTinHH 26](#_Toc311094028)

[2.2.2.5. XemTTHHTrongHD 27](#_Toc311094029)

[2.2.3. SellService: 27](#_Toc311094030)

[2.2.3.1. ThemHD 27](#_Toc311094031)

[2.2.3.2. SuaThongTinHD 27](#_Toc311094032)

[2.2.3.3. XemThongTinHD 28](#_Toc311094033)

[2.2.3.4. SuaTTHHTrongHD 28](#_Toc311094034)

[2.2.3.5. TongTienTrongHD 28](#_Toc311094035)

[2.2.4. SearchService: 28](#_Toc311094036)

[2.2.4.1. TimTVTheoTen 28](#_Toc311094037)

[2.2.4.2. TimNVTheoTen 29](#_Toc311094038)

[2.2.4.3. TimHHTheoTen 29](#_Toc311094039)

[2.2.4.4. LayDSHHTheoLoaiHang 29](#_Toc311094040)

[2.2.4.5. LayDSHHTheoDonGia 29](#_Toc311094041)

[2.2.5. TestService: 29](#_Toc311094042)

[2.2.5.1. KiemTraThanhVien 30](#_Toc311094043)

[2.2.5.2. KiemTraNhanVien 30](#_Toc311094044)

[2.2.5.3. KiemTraHangHoa 30](#_Toc311094045)

[2.2.5.4. KiemTraHoaDon 30](#_Toc311094046)

[2.2.6. OtherService: 30](#_Toc311094047)

[2.2.6.1. LayTenHH 30](#_Toc311094048)

[2.2.6.2. LayDonGia 31](#_Toc311094049)

[2.2.6.3. UpdateTongTien 31](#_Toc311094050)

[2.3. Mô tả CSDL: 31](#_Toc311094051)

[2.3.1. Các tables: 31](#_Toc311094052)

[2.3.1.1. NhanVien: 31](#_Toc311094053)

[2.3.1.2. HangHoa: 32](#_Toc311094054)

[2.3.1.3. ThanhVien: 32](#_Toc311094055)

[2.3.1.4. DSHHTrongHD: 32](#_Toc311094056)

[2.3.1.5. HoaDon: 33](#_Toc311094057)

[2.3.1.6. LoaiHang: 33](#_Toc311094058)

[3. Hiện thực: 34](#_Toc311094059)

[3.1. Các Form chính của chương trình: 34](#_Toc311094060)

[3.1.1. Người dùng quản lý: 34](#_Toc311094061)

[3.1.2. Người dùng nhân viên: 36](#_Toc311094062)

[3.2. Một số đoạn code đáng chú ý: 36](#_Toc311094063)

[3.2.1. Xử lý tìm kiếm nhân viên theo tên: 36](#_Toc311094064)

[3.2.2. Xử lý tính tổng tiền trong hóa đơn: 37](#_Toc311094065)

[3.2.3. Xử lý lấy danh sách hàng hóa theo loại hàng: 37](#_Toc311094066)

[3.3. Kết quả trả về và một số lỗi: 38](#_Toc311094067)

[3.3.1. Lỗi đăng nhập: 38](#_Toc311094068)

[3.3.1.1. Đăng nhập sai quyền hạn: 38](#_Toc311094069)

[3.3.1.2. Đăng nhập sai hoặc thiều thông tin: 38](#_Toc311094070)

[3.3.1.3. Đăng nhập thành công: 38](#_Toc311094071)

[3.3.2. Lỗi trùng khóa chính: 39](#_Toc311094072)

[3.3.2.1. Trùng mã hàng hóa: 39](#_Toc311094073)

[3.3.2.2. Trùng mã thành viên: 39](#_Toc311094074)

[3.3.2.3. Trùng mã nhân viên: 39](#_Toc311094075)

[3.3.3. Lỗi chưa kết nối đến Service: 39](#_Toc311094076)

[KẾT LUẬN 40](#_Toc311094077)

[TÀI LIỆU THAM KHẢO 41](#_Toc311094078)

# LỜI MỞ ĐẦU

Ngày nay việc quản lý siêu thị bằng máy tính đã trở nên rất quen thuộc với hầu hết chúng ta. Nó không những mang lại sự tiện lợi về thống kê, tính toán cho chủ doanh nghiệp mà còn mang lại sự nhanh chóng và tiện lợi trong việc thanh toán cho khách hàng.

Tuy nhiên, với hệ thống hiện nay, tùy vào mục đích người sử dụng mà ta phải viết các phần mềm sao cho phù hợp với yêu cầu người đó, chính vì vậy gây ra tốn kém về thời gian và tiền bạc. Do đó nhóm có ý tưởng viết một dịch vụ để quản lý siêu thị, bao gồm tất cả các chức năng mà tùy đối tượng ta có thể sử dụng.

Trong service bao gồm các chức năng:

1. Đối với quản lý:
   * Thêm, sửa, xóa, tìm kiếm nhân viên
   * Sửa số lượng mặt hàng trong hóa đơn
   * Thêm, sửa, xóa hàng hóa trong siêu thị
   * Thống kê tổng số lượng hàng bán được trong ngày
   * Thêm, sửa, xóa khách hàng thành viên
2. Đối với nhân viên:

* Xem danh sách mặt hàng
* Tạo hóa đơn
* Thêm mặt hàng vào hóa đơn
* Tìm kiếm sản phẩm theo mã sản phẩm, mã loại, đơn giá

1. Đối với khách hàng:

* Tìm vị trí mặt hàng trong siêu thị theo tên sản phẩm
* Hưỡng ưu đãi khi mua hàng thành viên
* Đi chợ ảo

**Các thành viên trong nhóm:**

1. Nguyễn Hoàng Duy ĐT:0988488070 Email: [hoangduy24061990@g.c](mailto:hoangduy24061990@g.c)
2. Lê Quang Hà
3. Tạ Ngọc Đại

# Tìm hiểu lý thuyết:

## Tổng quan về WCF:

WCF là công nghệ nền tảng nhằm thống nhất nhiều mô hình lập trình giao tiếp được hỗ trợ trong .NET 2.0 thành một mô hình duy nhất. Vào tháng 11 năm 2005, .NET 2.0 được Microsoft phát hành trong đó có cung cấp các hàm API riêng biệt cho các liên lạc dựa trên SOAP để tối đa hoá sự làm việc giữa các nền tảng sử dụng Web Services, đồng thời .NET 2.0 còn cung cấp các API để tối ưu việc liên lạc dựa trên mã nhị phân giữa các ứng dụng chạy trên hệ thống Windows gọi là .NET Remoting, các API cho các giao dịch phân tán, và API cho liên lạc dị bộ. WCF thống nhất các API này thành một mô hình duy nhất nhằm đáp ứng mô hình lập trình hướng dịch vụ.

WCF có thể sử dụng các bản tin SOAP giữa hai tiến trình, do đó làm cho các ứng dụng dựa trên WCF có thể làm việc với các tiến trình khác thông qua việc giao tiếp sử dụng bản tin SOAP. Khi một tiến trình WCF liên lạc với một tiến trình không là WCF, các bản tin SOAP được mã hoá trên cơ sở XML, nhưng khi nó liên lạc với một tiến trình WCF khác, bản tin SOAP có thể được tối ưu hoá dựa trên mã hoá nhị phân.

## Tại sao lại sử dụng WCF:

Như phần trên đã trình bày, .NET 2.0 hỗ trợ rất nhiều phương pháp liên lạc giữa các ứng dụng khác nhau nhằm vào các mục tiêu khác nhau. Các phương pháp liên lạc này khá phức tạp và phải mất nhiều thời gian để làm chủ được công nghệ. Tuy nhiên kiến thức thu được từ việc triển khai một phương pháp ít có khả năng dùng được khi làm việc với phương pháp khác.

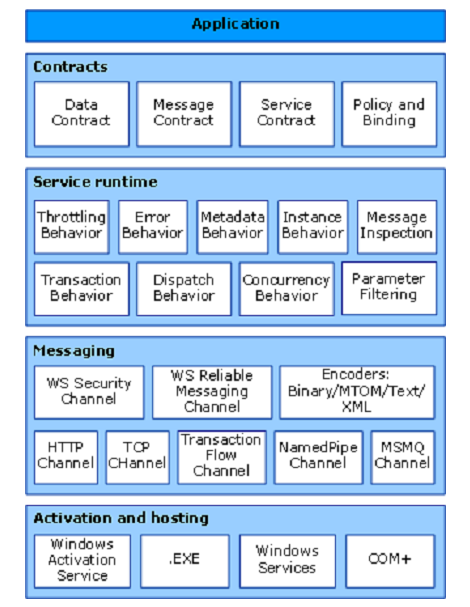
Với việc ra đời của WCF, mọi phương pháp liên lạc trước kia đều có thể thực hiện trên WCF. Do vậy nhà phát triển chỉ cần làm chủ được công nghệ WCF là có thể xây dựng các ứng dụng một cách nhanh chóng.

WCF là một mô hình lập trình cho phép nhà phát triển xây dựng các giải pháp dịch vụ đảm bảo tính ổn định, và bảo mật và thậm chí là đảm bảo giao dịch. Nó làm đơn giản hoá việc phát triển các ứng dụng nối kết và đưa ra cho nhà phát triển những giá trị mà có thể họ chưa nhận ra ngay, đó là cách tiếp cận phát triển hệ thống phân tán thống nhất, đơn giản, và quản lý được.

Do WCF được xây dựng trên cơ sở của .NET Framework 2.0 CLR, nó là tập các lớp cho phép các nhà phát triển xây dựng các ứng dụng hướng dịch vụ bằng môi trường lập trình quen thuộc của họ như VB.NET hay C#.

## Kiến trúc của WCF:

Hình sau mô tả các lớp chủ yếu trong kiến trúc của Windows Communication Foundation (WCF)

  
Hình 1: Kiến trúc của WCF

### Contracts:

Các contract trong WCF cũng giống như các hợp đồng/hiệp định mà bạn ký trong đời sống thật. Một hợp đồng bạn ký có thể chứa các thông tin như kiểu công việc bạn sẽ làm, và những thông tin mà bạn muốn đưa ra cho các bên khác. WCF contract cũng chứa các thông tin tương tự như vậy. Contract định nghĩa các đặc tả trong hệ thống bản tin.Thông thường có các loại contract sau:

* Service Contract (Contract dịch vụ)
* Data Contract (Contract dữ liệu)
* Message Contract (Contract bản tin)

#### Service Contract:

Một contract dịch vụ định nghĩa các hành động hoặc phương thức có ở điểm cuối

dịch vụ và được đưa ra để máy khách có thể sử dụng. Nó còn định nghĩa một cách cơ bản các phép trao đổi bản tin như việc bản tin sẽ thế nào trong yêu cầu/trả lời hay trong liên lạc đơn công hoặc song công.

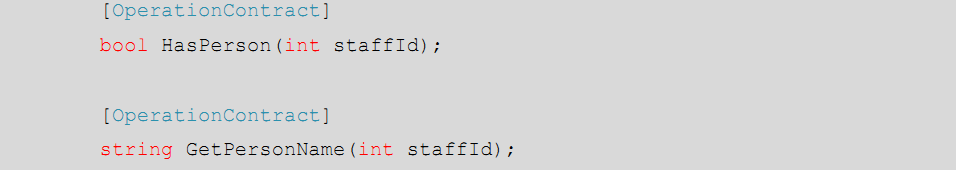
Contract dịch vụ đưa ra một số thông tin cho các máy khách đủ để cho máy khách có thể biết được dịch vụ này có thể cung cấp những gì. Những thông tin đó bao gồm:

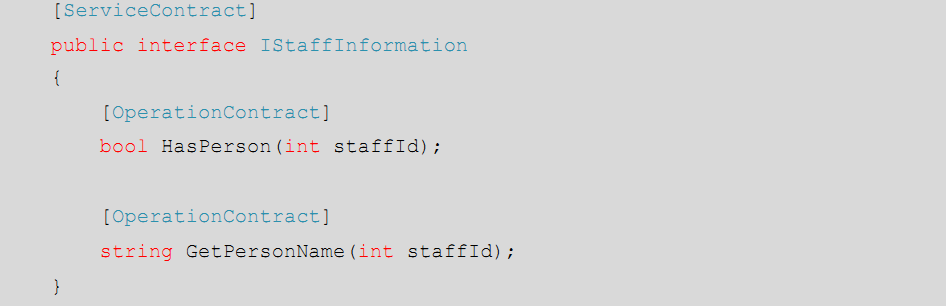
* Các kiểu dữ liệu trong bản tin
* Vị trí của các phương thức – hành động
* Thông tin về giao thức, định đạng lưu dữ liệu để đảm bảo cho liên lạc thành công
* Nhóm các hành động
* Mẫu trao đổi bản tin (Message exchange pattern – MEP)

Như trên đã nói, để định nghĩa contract dịch vụ, ta sử dụng các thuộc tính mô tả cho một lớp hay giao diện. Thuộc tính mô tả một contract dịch vụ là ServiceContract. Ví dụ sau định nghĩa một giao diện như là một contract dịch vụ:

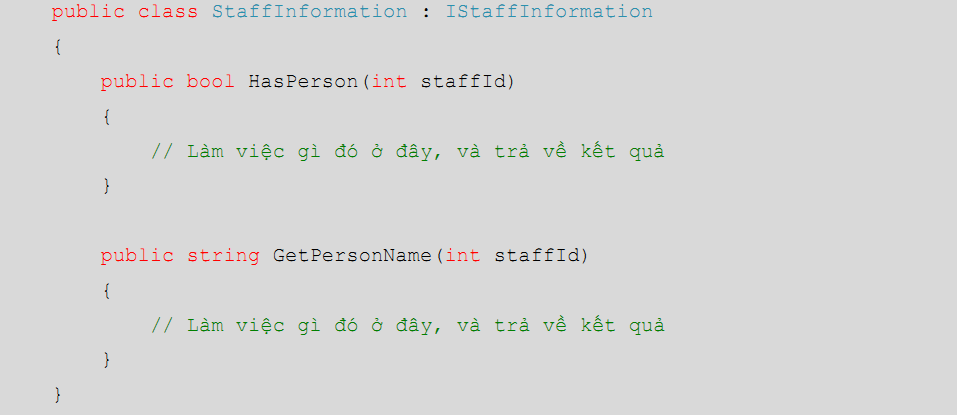


Sau đó để định nghĩa các hành động (phương thức) cho dịch vụ ta sử dụng phần mô tả là OperationContract cho các phương thức của giao diện như ví dụ dưới đây:

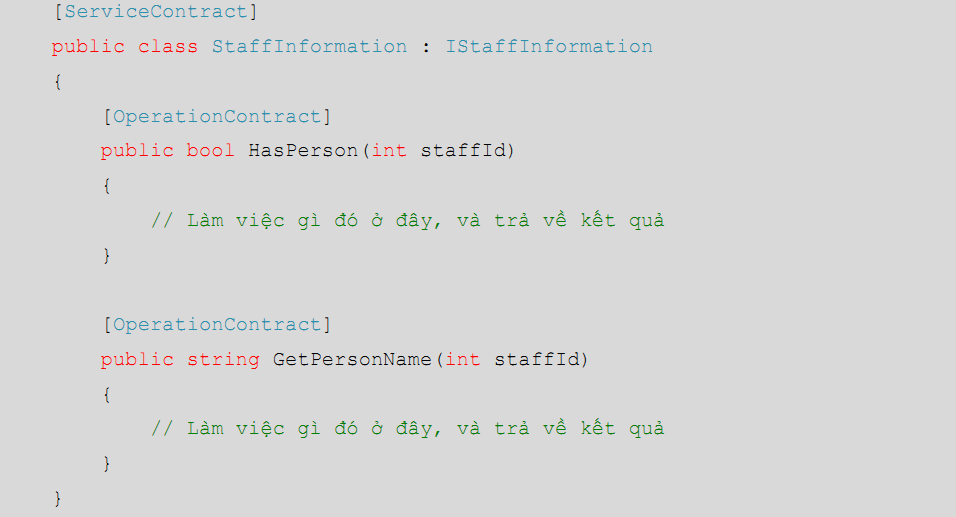
Gộp các phần lại ta sẽ được một định nghĩa hoàn chỉnh cho một contract dịch vụ:



Sau khi có được định nghĩa cho dịch vụ, ta có thể cài đặt một lớp cho giao diện trên, trong đó định nghĩa các hàm (phương thức) thực hiện theo logic của dịch vụ.



Trong trường hợp bạn không muốn định nghĩa dịch vụ ở phần giao diện, bạn hoàn toàn có thể định nghĩa dịch vụ ở trong lớp cài đặt dịch vụ đó như sau:



Trong khi định nghĩa contract dịch vụ, chúng ta đã sử dụng hai lớp thuộc tính để mô tả là ServiceContract và OperationContract. Hai thuộc tính này có rất nhiều tham số kèm theo, tuy nhiên trong nhiều trường hợp, chúng ta hoàn toàn có thể sử dụng các giá trị mặc định của hai thuộc tính này để định nghĩa contract dịch vụ. Trong một số trường hợp khi chúng ta muốn điều khiển sâu hơn về các thông tin liên quan đến dịch vụ, ta có thể đưa thêm các tham số cho hai thuộc tính này.

##### Thuộc tính Service Contract:

Thuộc tính ServiceContract được áp dụng cho việc mô tả các giao diện hoặc các lớp để định nghĩa một contract dịch vụ. Thuộc tính này có các tham số sau:





##### Thuộc tính Operation Contract:

Thuộc tính OperationContract được gắn với các phương thức trong các giao diện hay các lớp. Chỉ các phương thức được gắn thuộc tính OperationContract mới được coi là phương thức của dịch vụ. Các tham số có thể sử dụng cho thuộc tính này như sau:



#### Data Contract:

Một cách đơn giản thì một contract dữ liệu mô tả dữ liệu cần trao đổi. Trước khi máy khách và dịch vụ thực hiện liên lạc thì chúng phải đồng ý với nhau về kiểu dữ liệu trao đổi, đó là contract dữ liệu. Điều quan trọng là khi thực hiện liên lạc, máy khách và dịch vụ hoàn toàn không cần phải chung nhau cùng các kiểu dữ liệu mà chúng chỉ cần chung nhau các contract dữ liệu. Contract dữ liệu định nghĩa cách serialized và deserialized cho từng tham số và kiểu trả về.

Quá trình serialization dữ liệu là quá trình chuyển một cấu trúc dữ liệu thành một định dạng có thể dùng trong liên lạc hoặc gửi qua đường truyền. Ví dụ đọc dữ liệu từ cơ sở dữ liệu sau đó chuyển nó thành một chuỗi các byte và gửi qua đường truyền.

Quá trình deserialization là quá trình ngược lại với quá trình serialization. Quá trình này nhận dữ liệu từ đường truyền và chuyển ngược thành cấu trúc dữ liệu.

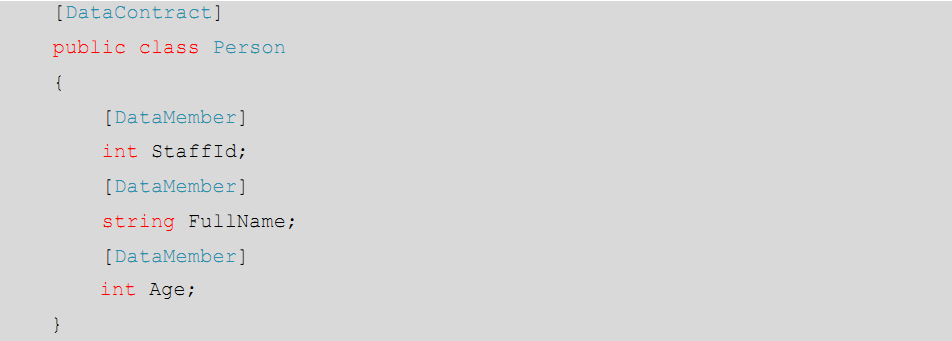
Để định nghĩa contract dữ liệu, ta sử dụng thuộc tính DataContract và DataMember.

##### Thuộc tính Data Contract:

Cũng giống như thuộc tính ServiceContract, thuộc tính DataContract được gắn vào các lớp để định nghĩa contract dữ liệu. Ví dụ sau định nghĩa một contract dữ liệu



Sau khi định nghĩa contract dữ liệu, bạn có thể định nghĩa các thành viên của một contract dữ liệu. Những thành viên này thực chất là các trường hay thuộc tính trong lớp của bạn. Ví dụ sau đây định nghĩa một số thành viên trong contract dữ liệu.



#### Message Contracts:

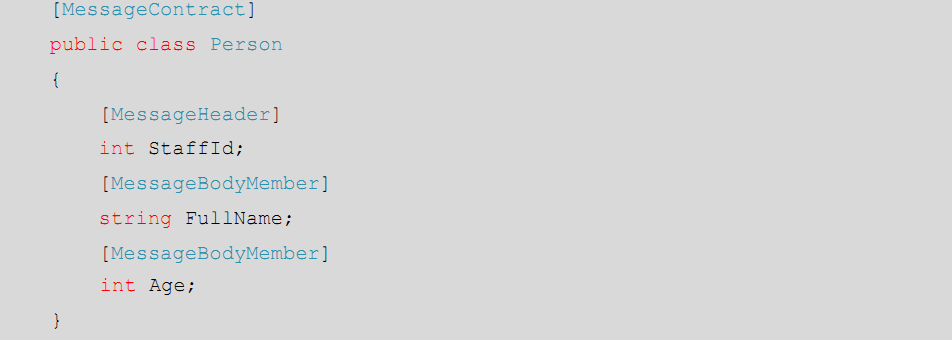
Contract bản tin cho phép bạn điều khiển toàn bộ định dạng của các bản tin SOAP. Trong phần lớn các trường hợp, bạn không cần sử dụng mức điều khiển này, do các contract dữ liệu đã cho phép bạn điều khiển theo yêu cầu. Tuy vậy, vẫn có một vài trường hợp bạn cần đến mức điều khiển bằng các contract bản tin.

Có lẽ lý do lớn nhất khiến bạn cần tới contract bản tin là tính làm việc liên môi trường. Một contract bản tin cung cấp các mức độ làm việc liên môi trường bạn cần khi liên lạc với các máy khách hoặc các hệ thống khác sử dụng một schema hay WSDL nào đó.

Contract bản tin cho phép bạn có nhiều lựa chọn hơn trong việc định dạng các tham số trong các bản tin SOAP ví như liệu thông tin có hay không có ở trong nội dung bản tin hay đầu đề bản tin. Để định nghĩa contract bản tin, ta sử dụng các lớp thuộc tính mô tả sau: MessageContract, MessageHeader, và MessageBodyMember cùng với các tham số tương ứng với từng lớp.

##### Thuộc tính MessageContract:

Các contract bản tin được định nghĩa bằng cách gắn thuộc tính MessageContract vào các lớp. Sau đó cần phải xác định rõ cho từng bản tin các thuộc tính MessageHeader và MessageBodyMember. Ví dụ sau biểu diễn một contract bản tin đơn giản.



##### Thuộc tính MessageHeader:

Thuộc tính MessageHeader được gắn với các trường hoặc thuộc tính của một kiểu. Những trường hay thuộc tính này sau đó sẽ được ánh xạ thành các đầu đề bản tin SOAP.

##### Thuộc tính MessageBodyMember:

Thuộc tính MessageBodyMember được gắn với các trường hoặc thuộc tính của một kiểu. Những trường hay thuộc tính này sau đó sẽ được ánh xạ thành phần nội dung của bản tin SOAP.

### Runtime service:

Lớp dịch vụ thực thi chứa các hành xử sẽ xảy ra trong quá trình thực hiện của dịch vụ, nghĩa là các hành xử thực thi của dịch vụ. Ta sẽ thấy một số các hành xử như sau:

* Throttling behavior: Điều khiển luồng nhằm quy định xem có bao nhiêu bản tin được xử lý
* Error behavior: Hành xử lỗi quy định những hành động khi lỗi xảy ra trong hệ thống
* Metadata behavior: Hành xử với các siêu dữ liệu quy định xem làm thế nào và khi nào thì các siêu dữ liệu được đưa ra bên ngoài dịch vụ
* Instance behavior: Hành xử thực thể quy định xem có bao nhiêu thực thể của dịch vụ đó được chạy
* Transaction behavior: Hành xử giao dịch cho phép việc rollback các giao dịch nếu xảy ra lỗi
* Message inspection: Kiểm tra bản tin đem lại cho dịch vụ khả năng kiểm tra tất cả hay một số phần của bản tin
* Dispatch behavior: Khi một bản tin được xử lý bởi nền tảng WCF, dịch vụ Dispatch behavior xác định xem bản tin được xử lý như thế nào.
* Concurrency behavior: Hành xử đồng thời xác định xem việc xử lý thế nào với việc đa luồng của mỗi dịch vụ hay mỗi thực thể của dịch vụ. Hành xử này giúp cho việc điều khiển số lượng luồng có thể truy nhập tới một thực thể của dịch vụ.
* Parameter filtering: Khi một bản tin được đưa tới một dịch vụ, sẽ xảy ra một số hành động dựa trên nội dung phần đầu đề của bản tin. Phần lọc tham số sẽ thực hiện lọc các đầu đề bản tin và thực hiện các hành động đặt sẵn dựa trên việc lọc đầu đề bản tin.

### Message:

Lớp bản tin là tập hợp các kênh. Mỗi kênh là một thành phần xử lý bản tin theo một cách nào đó. Một tập các kênh thường được gọi là ngăn xếp kênh. Các kênh làm việc trên bản tin và trên đầu đề của bản tin. Lớp này khác với lớp thực thi dịch vụ chủ yếu bởi sự khác nhau trong việc xử lý nội dung bản tin.

Có hai kênh khác nhau là kênh vận chuyển (transport channel) và kênh điều khiển (control channel).

* Kênh vận chuyển phụ trách việc đọc và ghi các bản tin từ mạng (network) hoặc từ một số điểm giao dịch bên ngoài)
* Kênh điều khiển thực hiện xử lý bản tin theo giao thức, thông thường làm việc bằng cách đọc và ghi thêm các đầu đề cho bản tin.

### Host and activation:

Nhìn một cách tổng thể thì một dịch vụ thực chất là một chương trình. Cũng giống như các chương trình khác, một dịch vụ cần phải chạy trong một tệp thực thi. Dịch vụ này thường được gọi là dịch vụ tự chứa.

Các dịch vụ còn có thể được chứa, hoặc chạy trong một tệp thực thi được quản lý bởi một agent bên ngoài như IIS hay Windows Activation Services (WAS). WAS cho phép WCF được kích hoạt một cách tự động khi phân phối tới một máy tính có chạy WAS.

## Mô hình lập trình WCF:

### Sử dụng phương pháp hướng đối tượng hay hướng dịch vụ?

Nếu các bạn đã học qua môn học về các phương pháp lập trình hẳn sẽ thấy có 2 phương pháp chính là hướng thủ tục và hướng đối tượng. Và phương pháp hướng đối tượng trong thời gian gần đây được phát triển rất mạnh và được hỗ trợ ngay trong các ngôn ngữ lập trình như C# hay VB.NET. Khi làm việc với các dịch vụ web bạn đã làm quen với một phương pháp nữa là lập trình hướng dịch vụ (Service-oriented programming). Microsoft đã cung cấp nhiều công cụ trên .NET Framework để hỗ trợ phương pháp lập trình này thông qua các lớp trong không gian tên: System.Web.Services.

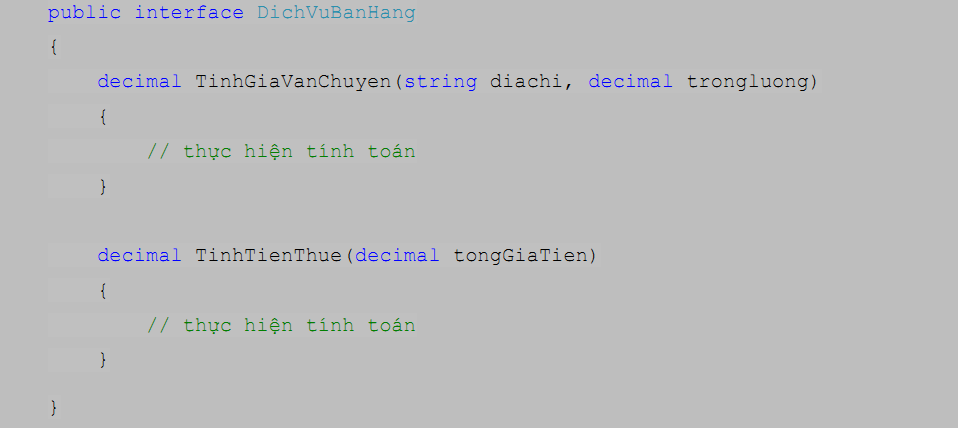
Như vậy khi làm việc với WCF bạn có hai lựa chọn là hướng đối tượng và hướng dịch vụ. Vậy ta nên sử dụng phương pháp nào? Câu trả lời là dùng cả hai. Nói một cách đơn giản là phương pháp hướng đối tượng được sử dụng để phát triển các ứng dụng trên desktop, còn phương pháp hướng dịch vụ được sử dụng để kết nối các ứng dụng đó với nhau. Điều quan trọng ở đây là làm sao để hiểu được sự khác nhau giữa hai phương pháp và hiểu được khi nào chúng được sử dụng và sử dụng như thế nào đồng thời cũng phải hiểu về các lợi ích chúng cung cấp.

Về hướng đối tượng có thể hiểu như sau. Các ứng dụng hướng đối tượng là hai hay nhiều lớp phụ thuộc lẫn nhau và chia sẻ chung các kiểu dữ liệu. Những lớp này liên lạc với nhau thông qua các lời gọi các hàm mà lớp đối tượng cung cấp.

Các ứng dụng hướng dịch vụ là các chương trình không biết gì về nhau. Mỗi ứng dụng liên lạc với ứng dụng khác thông qua các bản tin. Điểm đặc biệt là các bản tin này được gửi từ một ứng dụng sang ứng dụng khác mà không quan tâm tới nền tảng mà dịch vụ đang chạy.

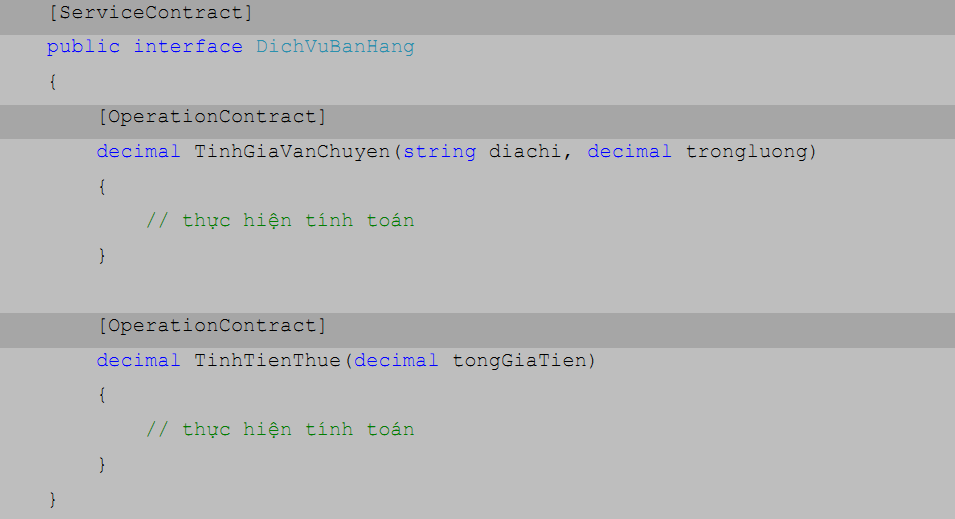
Khi phát triển các dịch vụ WCF, điều quan trọng là cần hiểu sự liên kết giữa hướng đối tượng và hướng dịch vụ. Khi làm việc với .NET Framework bạn chắc chắn rất quen thuộc với thuật ngữ lớp (class) và giao diện (interface). Các thuật ngữ này vẫn được sử dụng khi phát triển dịch vụ WCF. Các lớp và giao diện là phần hướng đối tượng trong WCF, còn phần hướng dịch vụ trong WCF sẽ được thấy khi bạn đưa vào các thuộc tính WCF để định nghĩa các thực thể.

Ví dụ, lớp sau đây định nghĩa một giao diện hướng đối tượng:



Hình 2: Định nghĩa một giao diện hướng đối tượng

Ta sẽ có phần hướng dịch vụ cho dịch vụ WCF khi thêm vào các thuộc tính cho giao diện ở trên:



Hình 3: Thêm thuộc tính cho giao diện theo hướng dịch vụ

Như vậy là qua ví dụ trên các bạn có thể thấy mối liên kết giữa phương pháp hướng đối tượng và hướng dịch vụ trong WCF. Các bạn chưa cần quan tâm tới các thuộc tính [ServiceContract] và [OperationContract] vội, bởi vì những thuộc tính này, và còn nhiều thứ khác nữa sẽ được giới thiệu một cách chi tiết khi thích hợp.

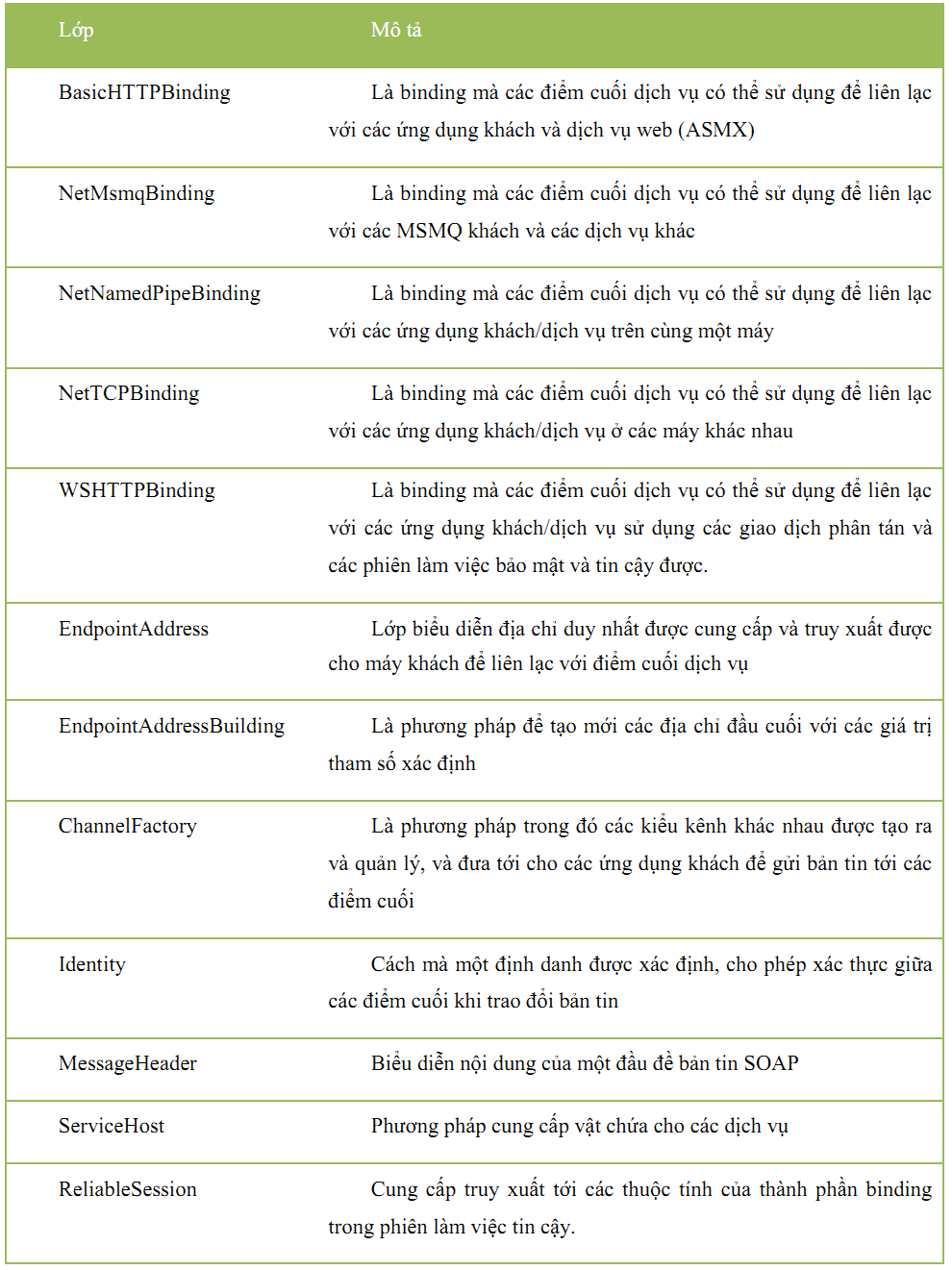
### Service Model (Mô hình dịch vụ)

Nếu bạn đã từng làm việc với dịch vụ web, bạn sẽ thấy mô hình này quen thuộc với bạn theo một cách nào đó. Khi bạn tạo một dịch vụ web, bạn thực sự tạo ra một dịch vụ (service). Dịch vụ web chứa một tài liệu XML để mô tả tất cả mọi thứ cần biết về dịch vụ đó. Tài liệu này được mô tả bằng ngôn ngữ Web Service Description Language (ngôn ngữ mô tả dịch vụ web). Nó chứa ba phần:

* Service (dịch vụ): Chứa thông tin về vị trí của dịch vụ
* Binding: Chứa thông tin về cách liên lạc với dịch vụ, như dịch vụ sử dụng giao thức gì, vv.
* PortType (kiểu cổng): Giải thích về dịch vụ sẽ làm gì

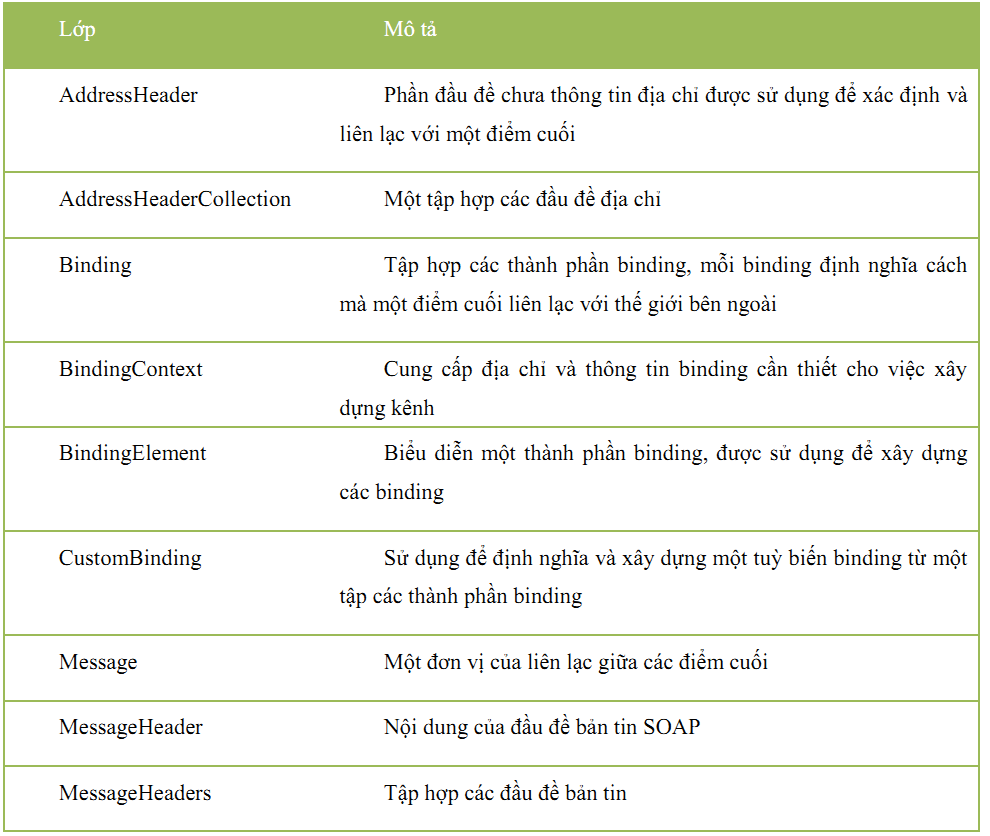
Mô hình dịch vụ trên WCF cũng tương tự như với mô hình dịch vụ web. Điểm khác biệt là ở cách đặt tên. Trong WCF các phần không được gọi là service, binding, và portType mà được gọi tương ứng là address (địa chỉ), binding, và contract.

Mô hình dịch vụ WCF được cung cấp trong không gian tên System.ServiceModel. Không gian tên này chứa rất nhiều lớp, nhưng bạn hoàn toàn không cần biết toàn bộ chúng. Để sử dụng mô hình và xây dựng dịch vụ, ta thường sử dụng một số lớp sau:



Hình 4: Các lớp trong Service Model

Để định nghĩa việc liên lạc của dịch vụ, ta thường hay sử dụng các lớp sau:



Hình 5: Các lớp để việc liên lạc của dịch vụ

### Các phương pháp lập trình với WCF:

Có một số phương pháp lập trình với WCF, mỗi phương pháp có ưu điểm và khuyết điểm riêng của nó. Điều đặc biệt về WCF là luôn có hơn một cách để giải quyết một vấn đề trong WCF, và bạn không nhất thiết phải chọn duy nhất một phương pháp nào. Trong thực tế, cách làm tốt nhất là tổ hợp các phương pháp để có được sự linh hoạt và mềm dẻo cho dịch vụ của bạn.

Có ba phương pháp hay được sử dụng khi phát triển dịch vụ WCF như sau:

* Phương pháp khai báo
* Phương pháp lập trình trực tiếp
* Phương pháp sử dụng tập tin cấu hình

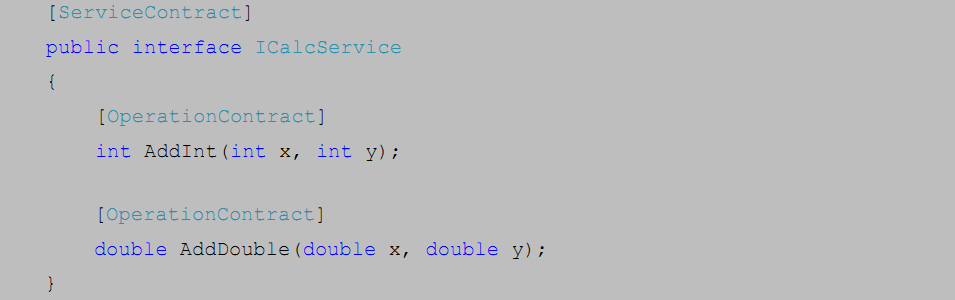
#### Declarative programming (Phương pháp khai báo)

Lập trình khai báo đạt được thông qua các thuộc tính. Những thuộc tính này được sử dụng để định nghĩa các contract và xác định hành xử của dịch vụ. Chúng được sử dụng để xác định thêm các tham số để thay đổi các chi tiết của contract và hành xử dịch vụ.

Thuộc tính ServiceContract dùng để quy định là giao diện này định nghĩa các chức năng của một dịch vụ. Thuộc tính OperationContract được sử dụng ở các hàm để quy định rằng hàm này được khai báo là một phần của dịch vụ. Đó là tất cả những gì cần để tạo ra một dịch vụ WCF.

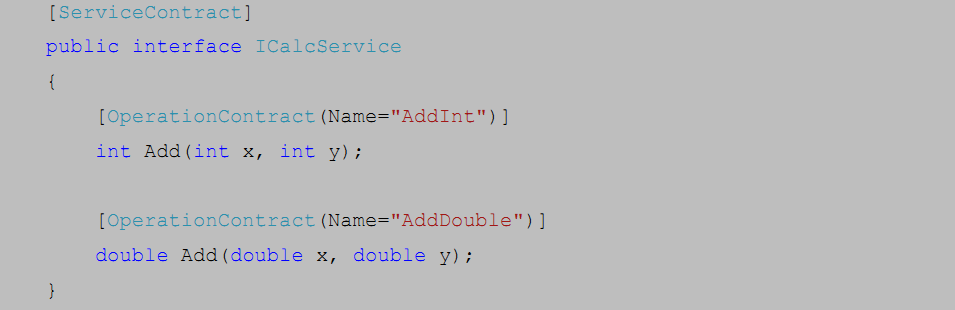
Thêm nữa, bạn không nhất thiết phải sử dụng các giao diện (interface) khi cài đặt một dịch vụ, điều này cũng giống như việc bạn không cần phải sử dụng giao diện để định nghĩa một lớp. Tuy vậy bạn nhất thiết phải quy định phần nào thuộc về dịch vụ. Bạn có thể định nghĩa những phần khác cần cho giao diện, nhưng chỉ những hàm (phương thức) có gắn thuộc tính [OperationContract].

Ví dụ ta có một dịch vụ thực hiện phép tính cộng giữa 2 số nguyên AddInt và 2 số thực AddDouble. Ta khai báo dịch vụ như sau:



Hình 6: Service thực thi phép cộng giữa 2 số nguyên và 2 số thực

Như vậy dịch vụ của chúng ta sau khi khai báo sẽ có 2 phương thức (khai báo với thuộc tính OperationContract) là AddInt và AddDouble. Tuy nhiên khi khai báo trong C#, việc đặt tên AddInt và AddDouble, và có thể có một số hàm add cho các kiểu dữ liệu khác, có thể rút gọn lại thành một tên hàm Add mà thôi. Nhưng các dịch vụ lại không cho phép đặt trùng tên hàm như thế. Chúng ta có thể khai báo thêm với thuộc tính OperationContract để thực hiện, cách làm như sau:



Hình 7:Khai báo thêm thuộc tính Name cho 2 Contract

Các bạn có thể thấy là chúng ta sử dụng được phép nạp chồng tên trong C# và sử dụng thêm tham số Name để quy định thêm tên hàm ở dịch vụ. Ngoài ưu điểm trong việc giải quyết nạp chồng tên hàm, ta còn thấy một lợi ích khác nữa là, việc quy định tham số Name trong thuộc tính OperationContract còn cho ta thêm linh hoạt trong việc đổi tên các hàm trong giao diện mà không làm thay đổi định nghĩa dịch vụ, nghĩa là các ứng dụng khác sử dụng dịch vụ này không cần phải biên dịch lại.

#### Explicit programming (Phương pháp lập trình trực tiếp)

Là phương pháp lập trình hướng đối tượng, bạn làm việc trực tiếp với các lớp và giao diện cung cấp bởi mô hình đối tượng của WCF. Làm việc trực tiếp với mô hình đối tượng cho phép nhà phát triển tính linh hoạt cao hơn và khả năng điều khiển tốt hơn thông qua mã nguồn của họ. Thêm nữa nó cho phép điều khiển sâu hơn rất nhiều so với phương pháp khai báo và phương pháp sử dụng tập tin cấu hình.

#### Phương pháp sử dụng tập tin cấu hình

Cũng giống như phương pháp khai báo, có rất nhiều thứ mà bạn có thể quy định liên quan đến hành xử của một dịch vụ thông qua tập tin cấu hình của dịch vụ. Điều hay trong cách tiếp cận này là những thay đổi ở tập tin cấu hình hoàn toàn không cần phải biên dịch lại dịch vụ mới sử dụng được.

Sau đây là ví dụ sử dụng tập tin cấu hình để định nghĩa dịch vụ tính toán trong ví dụ của phần phương pháp khai báo.



Hình 8: Một file cấu hình (app.config) mẫu

# Thiết kế chương trình:

## Mô tả chung:

### Tổng quan về chương trình:

Service quản lý siêu thị được viết ra nhằm đáp ứng các yêu cầu phục vụ cho nhân viên bán hàng trong siêu thị, hỗ trợ khách hàng và quản lý của siêu thị.

Nó cung cấp các dịch vụ cơ bản như: quản lý nhân viên, nhập hàng vào siêu thị, xuất hóa đơn bán hàng cho khách hàng. Ngoài ra còn có các tính năng nâng cao như tìm kiếm mặt hàng, vị trí của mặt hàng, giá cả của mặt hàng phục vụ mua sắm của khách.

### Chức năng và đối tượng sử dụng chương trình:

1. Quản lý:
   * Thêm nhân viên
   * Sửa thông tin nhân viên
   * Tìm kiếm nhân viên dựa vào tên nhân viên
   * Sửa số lượng mặt hàng trong hóa đơn
   * Sửa thông tin mặt hàng trong hóa đơn
   * Nhập hàng hóa vào siêu thị
   * Xóa mặt hàng trong siêu thị
   * Sửa thông tin hàng hóa
   * Thống kê tổng số lượng hàng bán được trong ngày
   * Quản lý khách hàng thành viên
2. Nhân viên:

* Xem danh sách mặt hàng
* Tạo hóa đơn
* Thêm mặt hàng vào hóa đơn
* Tìm kiếm sản phẩm theo mã sản phẩm
* Tìm kiếm sản phẩm theo mã loại
* Tìm kiếm sản phẩm theo đơn giá

1. Khách hàng:

* Tìm vị trí mặt hàng trong siêu thị theo tên sản phẩm
* Hưỡng ưu đãi khi mua hàng nếu là GoldMember
* Hưỡng ưu đãi khi mua hàng nếu là SilverMember
* Đi chợ ảo

## Mô tả chi tiết:

### PersonService:

* Chức năng: cung cấp thông tin về nhân viên, quản lý và khách hàng thành viên trong siêu thị
* Tình huống dịch vụ: được sử dụng khi người dùng với quyền quản lý muốn thêm, sửa, xóa, xem thông tin cá nhân của các nhân viên, quản lý và khách hàng thành viên
* Các contract:

#### ThemNV

* Chức năng: thêm thông tin một nhân viên
* Input: class NhanVien
* Output: thêm thông tin nhân viên mới vào csdl
* Binding: basicHttpBinding.

#### XoaNV

* Chức năng: xóa thông tin một nhân viên dựa vào MaNV
* Input: MaNV(string)
* Output: xóa thông tin của nhân viên đó trong csdl
* Binding: basicHttpBinding.

#### SuaThongTinNV

* Chức năng: sửa thông tin một nhân viên dựa vào MaNV
* Input: MaNV(string)
* Output: sửa thông tin của nhân viên đó trong csdl
* Binding: basicHttpBinding.

#### XemThongTinNV

* Chức năng: xem thông tin của toàn bộ danh sách nhân viên
* Input: null
* Output: hiển thị thông tin của toàn bộ danh sách nhân viên được biểu diễn ở dạng List<NhanVien> có trong siêu thị được sắp xếp tăng dần theo MaNV
* Binding: basicHttpBinding.

#### ThemTV

* Chức năng: thêm thông tin một thành viên
* Input: class ThanhVien
* Output: thêm thông tin thành viên mới vào csdl
* Binding: basicHttpBinding.

#### XoaTV

* Chức năng: xóa thông tin một thành viên dựa vào MaNT
* Input: MaTV(string)
* Output: xóa thông tin của thành viên đó trong csdl
* Binding: basicHttpBinding.

#### SuaThongTinTV

* Chức năng: sửa thông tin một thành viên dựa vào MaTV
* Input: MaTV(string)
* Output: sửa thông tin của thành viên đó trong csdl
* Binding: basicHttpBinding.

#### XemThongTinTV

* Chức năng: xem thông tin của toàn bộ danh sách thành viên
* Input: null
* Output: hiển thị thông tin của toàn bộ danh sách thành viên được biểu diễn ở dạng List<ThanhVien> của siêu thị được sắp xếp tăng dần theo MaTV
* Binding: basicHttpBinding.

### ProductService:

* Chức năng: cung cấp thông tin về hàng hóa trong siêu thị
* Tình huống dịch vụ: được sử dụng khi người dùng (quản lý và nhân viên) muốn thêm, sửa, xóa, xem thông tin hàng hóa có trong siêu thị
* Các contract:

#### ThemHH

* Chức năng: thêm một sản phẩm mới vào siêu thị
* Input: class HangHoa
* Output: thêm thông tin sản phẩm mới vào csdl
* Binding: basicHttpBinding.

#### XoaHH

* Chức năng: xóa đi một sản phẩm dựa vào MaHH
* Input: MaHH(string)
* Output: xóa sản phẩm đó trong csdl
* Binding: basicHttpBinding.

#### SuaThongTinHH

* Chức năng: sửa thông tin của sản phẩm dựa vào MaHH
* Input: MaHH(string)
* Output: sửa thông tin của hàng hóa đó trong csdl
* Binding: basicHttpBinding.

#### XemThongTinHH

* Chức năng: xem thông tin của toàn bộ danh sách hàng hóa
* Input: null
* Output: hiển thị thông tin của toàn bộ danh sách hàng hóa được biểu diễn ở dạng List<HangHoa> có trong siêu thị được sắp xếp tăng dần theo MaHH
* Binding: basicHttpBinding.

#### XemTTHHTrongHD

* Chức năng: xem thông tin của toàn bộ hàng hóa có trong hóa đơn theo MaHD
* Input: MaHD
* Output: hiển thị thông tin của toàn bộ hàng hóa có trong hóa đơn đó được biểu diễn ở dạng List<DSHHTrongHD> và được sắp xếp tăng dần theo MaHH
* Binding: basicHttpBinding.

### SellService:

* Chức năng: cung cấp thông tin về quá trình thanh toán hóa đơn trong siêu thị
* Tình huống dịch vụ: được sử dụng khi người dùng (quản lý và nhân viên) thanh toán tiền cho khách hàng
* Các contract:

#### ThemHD

* Chức năng: thêm một hóa đơn mới
* Input: class HoaDon
* Output: thêm thông tin hóa đơn mới vào csdl
* Binding: basicHttpBinding.

#### SuaThongTinHD

* Chức năng: quản lý sửa thông tin của hóa đơn dựa vào MaHD
* Input: MaHD(string)
* Output: sửa thông tin của hóa đơn đó trong csdl
* Binding: basicHttpBinding.

#### XemThongTinHD

* Chức năng: xem thông tin của toàn bộ danh sách hóa đơn
* Input: null
* Output: hiển thị thông tin của toàn bộ danh sách hóa đơn được biểu diễn ở dạng List<HoaDon> có trong siêu thị được sắp xếp tăng dần theo MaHD
* Binding: basicHttpBinding.

#### SuaTTHHTrongHD

* Chức năng: quản lý sửa thông tin của các mặt hàng có trong hóa đơn dựa vào MaHD
* Input: MaHD(string)
* Output: sửa thông tin sản phẩm trong hóa đơn đó trong csdl
* Binding: basicHttpBinding.

#### TongTienTrongHD

* Chức năng: tính tổng số tiền có trong hóa đơn theo MaHD
* Input: MaHD(string)
* Output: tính ra tổng số tiền có trong hóa đơn đó
* Binding: basicHttpBinding.

### SearchService:

* Chức năng: cung cấp các chức năng tìm kiếm sản phẩm, nhân viên, thành viên; liệt kê danh sách hàng hóa theo tên, giá…
* Tình huống dịch vụ: được sử dụng khi người dùng tìm kiếm sản phẩm, nhân viên, thành viên theo mã, tên, đơn giá…
* Các contract:

#### TimTVTheoTen

* Chức năng: tìm và đưa ra thông tin của thành viên (nếu có) dựa theo TenTV
* Input: TenTV
* Output: thông tin của thành viên được biểu diện ở dạng List<ThanhVien>
* Binding: basicHttpBinding.

#### TimNVTheoTen

* Chức năng: liệt kê thông tin các mặt hàng dựa theo HoTenNV
* Input: HoTenNV(string)
* Output: hiển thị thông tin hàng hóa được biểu diện ở dạng List<NhanVien>
* Binding: basicHttpBinding.

#### TimHHTheoTen

* Chức năng: liệt kê thông tin các mặt hàng dựa theo TenHang
* Input: TenHang(string)
* Output: hiển thị thông tin hàng hóa được biểu diện ở dạng List<HangHoa>
* Binding: basicHttpBinding.

#### LayDSHHTheoLoaiHang

* Chức năng: liệt kê thông tin các mặt hàng dựa theo MaLoai
* Input: MaLoai(string)
* Output: hiển thị thông tin hàng hóa được biểu diện ở dạng List<HangHoa>
* Binding: basicHttpBinding.

#### LayDSHHTheoDonGia

* Chức năng: liệt kê thông tin các mặt hàng dựa theo DonGia
* Input: DonGia(string)
* Output: hiển thị thông tin hàng hóa được biểu diện ở dạng List<HangHoa>
* Binding: basicHttpBinding.

### TestService:

* Chức năng: kiểm tra xem mã nhân viên, thành viên, sản phẩm, hóa đơn có tồn tại hay không
* Tình huống dịch vụ: được sử dụng khi người dùng tạo mới nhân viên, thành viên, hàng hóa hoặc sản phẩm, kiểm tra xem thông tin có bị trùng hay không
* Các contract:

#### KiemTraThanhVien

* Chức năng: kiểm tra xem MaTV có tồn tại hay chưa
* Input: MaTV(string)
* Output: nếu tồn tại thì trả về giá trị bool = true và ngược lại
* Binding: basicHttpBinding.

#### KiemTraNhanVien

* Chức năng: kiểm tra xem MaNV có tồn tại hay chưa
* Input: MaNV(string)
* Output: nếu tồn tại thì trả về giá trị bool = true và ngược lại
* Binding: basicHttpBinding.

#### KiemTraHangHoa

* Chức năng: kiểm tra xem MaHH có tồn tại hay chưa
* Input: MaHH(string)
* Output: nếu tồn tại thì trả về giá trị bool = true và ngược lại
* Binding: basicHttpBinding.

#### KiemTraHoaDon

* Chức năng: kiểm tra xem MaHD có tồn tại hay chưa
* Input: MaHD(string)
* Output: nếu tồn tại thì trả về giá trị bool = true và ngược lại
* Binding: basicHttpBinding.

### OtherService:

* Chức năng: cung cấp một số chức năng khác cho người dùng
* Tình huống dịch vụ: được sử dụng để hỗ trợ cho các Service khác
* Các contract:

#### LayTenHH

* Chức năng: Lấy tên hàng hóa theo mã hàng
* Input: MaHH(string)
* Output: hiển thị tên sản phẩm cần xem
* Binding: basicHttpBinding.

#### LayDonGia

* Chức năng: Lấy đơn giá theo mã hàng
* Input: MaHH(string)
* Output: hiển thị đơn giá của sản phẩm cần xem
* Binding: basicHttpBinding.

#### UpdateTongTien

* Chức năng: cập nhật tổng số tiền trong hóa đơn mỗi khi nhập thêm hàng hóa vào hóa đơn
* Input: MaHH(string), MaHD(string)
* Output: cập nhật tổng tiển trong hóa đơn

## Mô tả CSDL:

### Các tables:

#### NhanVien:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| MaNV | varchar(20) | No | Khóa chính |
| MatKhau | varchar(50) | Yes | Mật khẩu đăng nhập |
| HoTenNV | nvarchar(100) | Yes | Họ tên nhân viên |
| NgaySinh | date | Yes | Ngày sinh |
| NoiSinh | nvarchar(50) | Yes | Nơi sinh |
| DiaChi | nvarchar(200) | Yes | Địa chỉ |
| SoDienThoai | varchar(20) | Yes | Số điện thoại |

#### HangHoa:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| MaHH | varchar(50) | No | Khóa chính |
| TenHH | nvarchar(100) | Yes | Tên sản phẩm |
| DonGia | money | Yes | Đơn giá |
| SoLuong | int | Yes | Số lượng |
| MaLoai | varchar(10) | No | Mã loại |
| ViTri | nvarchar(100) | Yes | Vị trí |

#### ThanhVien:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| MaTV | varchar(100) | No | Khóa chính |
| TenTV | nvarchar(100) | Yes | Họ tên thành viên |
| DiaChi | nvarchar(100) | Yes | Địa chỉ |
| SoDienThoai | varchar(20) | Yes | Số điện thoại |
| CMND | varchar(20) | Yes | Số CMND |
| DiemTichLuy | int | Yes | Điểm tích lũy |
| LoaiTV | nchar(10) | Yes | Loại thành viên |

#### DSHHTrongHD:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| MaHD | varchar(100) | No | Khóa chính |
| MaHH | varchar(50) | Yes | Họ tên thành viên |
| TenHH | nvarchar(100) | Yes | Địa chỉ |
| DonGia | money | Yes | Số điện thoại |
| SoLuongMua | int | Yes | Số CMND |
| ThanhTien | money | Yes | Điểm tích lũy |

#### HoaDon:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| MaHD | varchar(100) | No | Khóa chính |
| NgayXuatHD | date | Yes | Họ tên thành viên |
| TongTien | money | Yes | Địa chỉ |

#### LoaiHang:

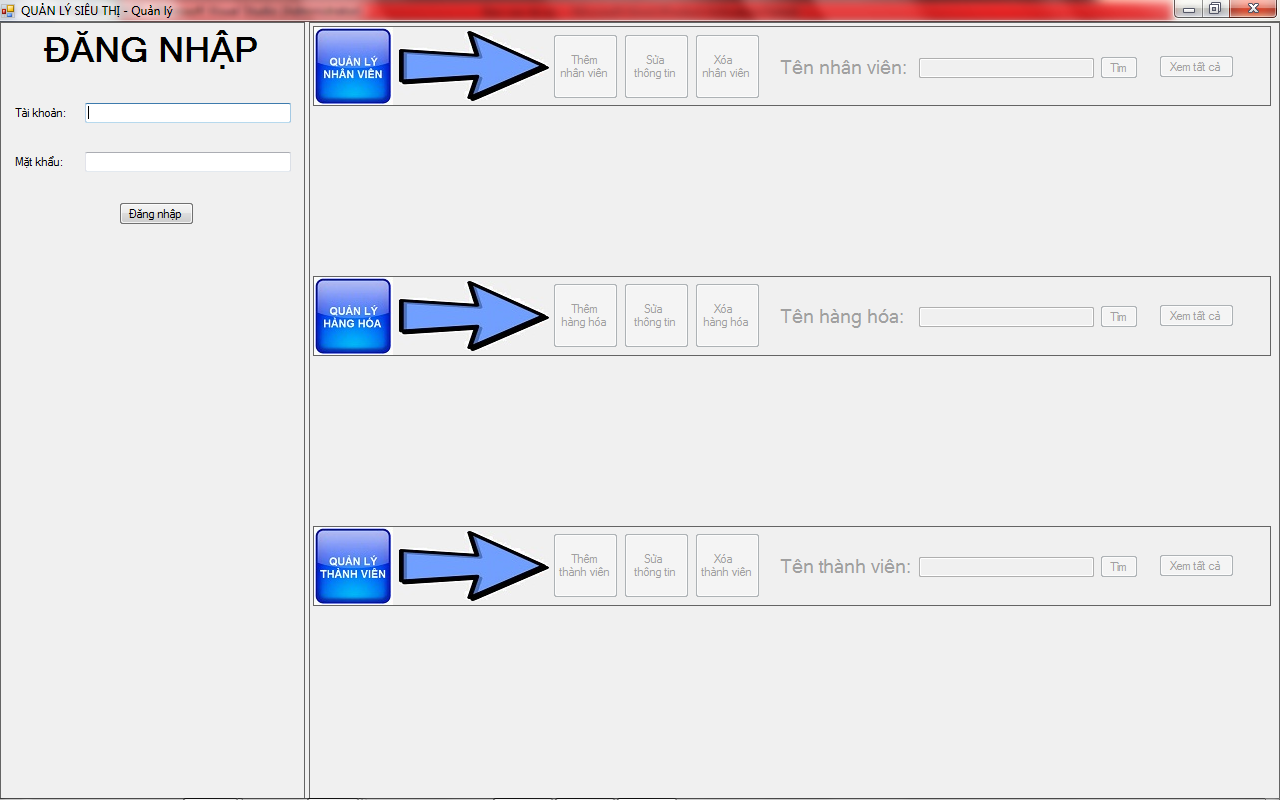
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value Type | Null | Description |
| MaLoai | varchar(10) | Yes | Họ tên thành viên |
| TenLoai | nvarchar(100) | Yes | Địa chỉ |

# Hiện thực:

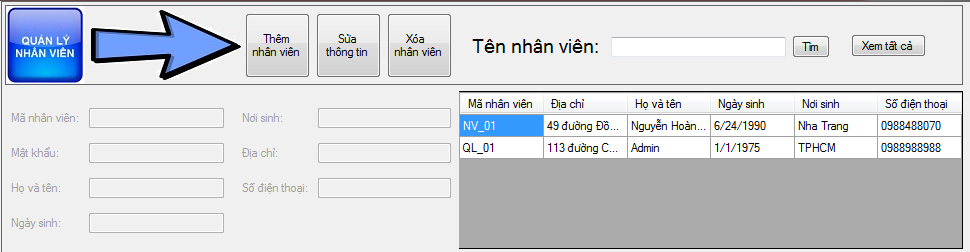
## Các Form chính của chương trình:

### Người dùng quản lý:

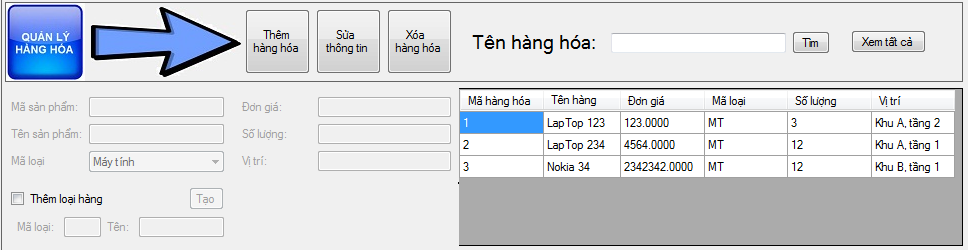
* Trước khi đăng nhập:



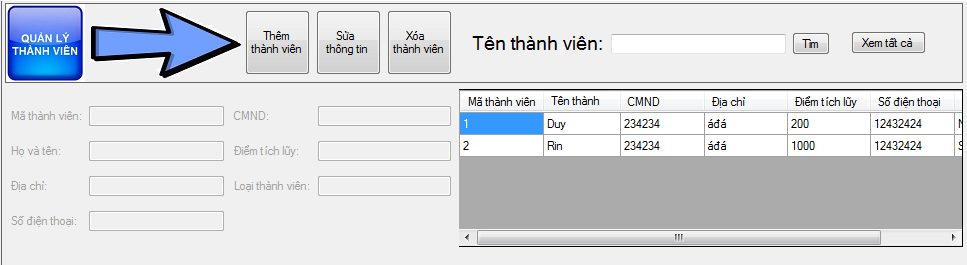
* Sau khi đăng nhập:
  + Quản lý nhân viên:



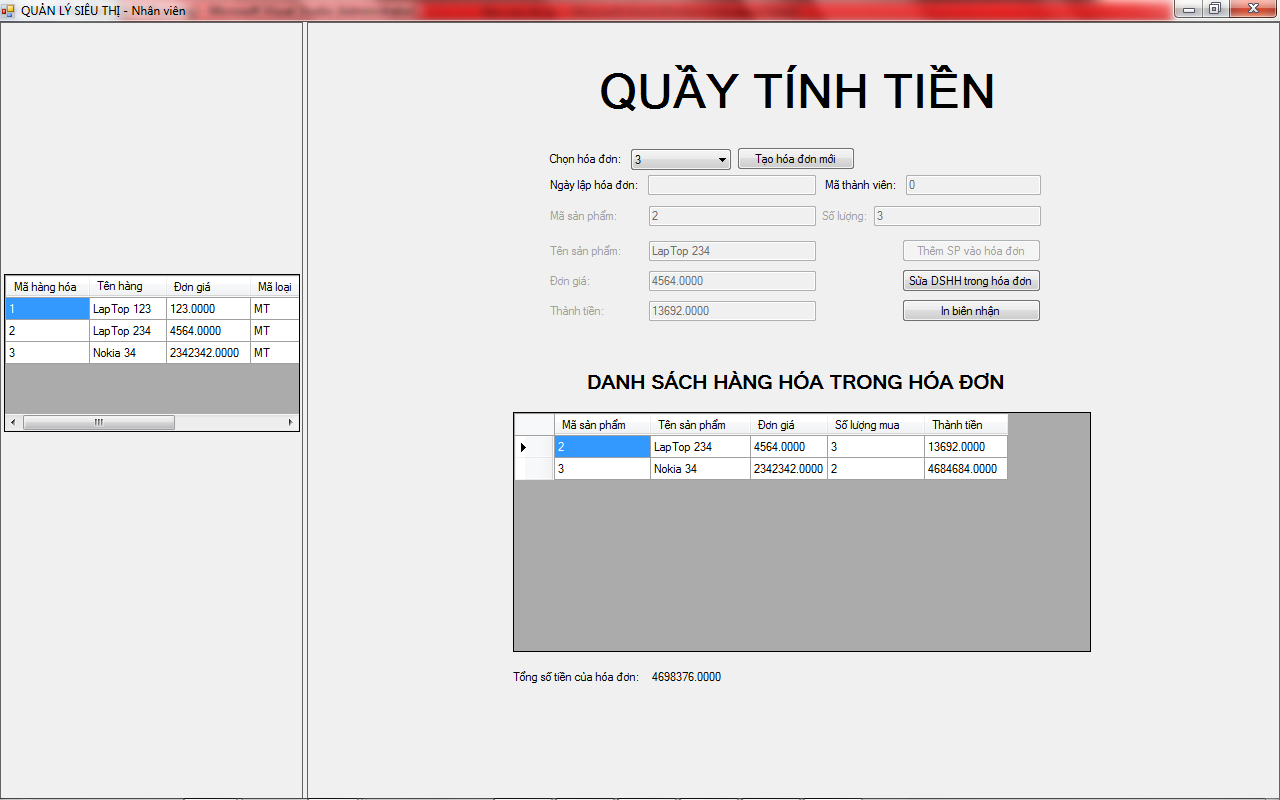
* + Quản lý hàng hóa:



* + Quản lý thành viên:



### Người dùng nhân viên:



## Một số đoạn code đáng chú ý:

### Xử lý tìm kiếm nhân viên theo tên:

public NhanVien[] TimNVTheoTen(string TenNV)

{

List<NhanVien> dsnv = new List<NhanVien>();

string SQLStr = "select \* from NhanVien where HoTenNV like N'%" + TenNV + "%'";

SqlConnection con = new SqlConnection(ConnectionStr);

SqlCommand cmd = new SqlCommand(SQLStr, con);

con.Open();

SqlDataReader dr = cmd.ExecuteReader();

if (dr.HasRows)

{

while (dr.Read())

{

NhanVien nv = new NhanVien();

nv.MaNV = dr.GetString(0);

nv.MatKhau = dr.GetString(1);

nv.HoTenNV = dr.GetString(2);

nv.NgaySinh = dr.GetDateTime(3);

nv.NoiSinh = dr.GetString(4);

nv.DiaChi = dr.GetString(5);

nv.SoDienThoai = dr.GetString(6);

dsnv.Add(nv);

}

}

return dsnv.ToArray();

}

### Xử lý tính tổng tiền trong hóa đơn:

public decimal TongTienTrongHD(string MaHD)

{

decimal kq = 0;

string SQLStr = "select ThanhTien from DSHHTrongHD where MaHD = " + MaHD + "";

SqlConnection conn = new SqlConnection(ConnectionStr);

SqlCommand cmd = new SqlCommand(SQLStr, conn);

conn.Open();

SqlDataReader rd = cmd.ExecuteReader();

if (rd.HasRows)

while(rd.Read())

kq += rd.GetDecimal(0);

return kq;

}

### Xử lý lấy danh sách hàng hóa theo loại hàng:

public HangHoa[] LayDSHHTheoLoaiHang(string MaLoai)

{

//Khởi tạo câu lệnh SQL lấy DSHH

List<HangHoa> DSSanPham = new List<HangHoa>();

string SQLStr = "select \* from HangHoa where MaLoai = '" + MaLoai + "'";

SqlConnection con = new SqlConnection(ConnectionStr);

SqlCommand cmd = new SqlCommand(SQLStr, con);

con.Open();

SqlDataReader dr = cmd.ExecuteReader();

if (dr.HasRows)

{

while (dr.Read())

{

HangHoa hh = new HangHoa();

hh.MaHH = dr.GetString(0);

hh.TenHH = dr.GetString(1);

hh.DonGia = dr.GetDecimal(2);

hh.SoLuong = dr.GetInt32(3);

hh.MaLoai = dr.GetString(4);

hh.ViTri = dr.GetString(5);

//Thêm vào danh sách

DSSanPham.Add(hh);

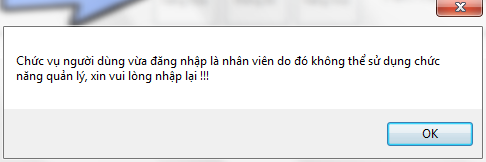
}

}

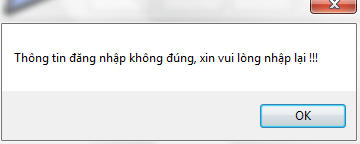
## Kết quả trả về và một số lỗi:

### Lỗi đăng nhập:

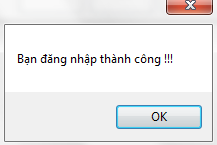
#### Đăng nhập sai quyền hạn:



#### Đăng nhập sai hoặc thiều thông tin:

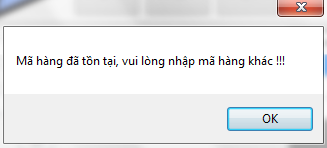


#### Đăng nhập thành công:

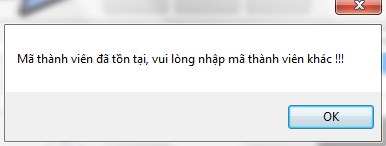


### Lỗi trùng khóa chính:

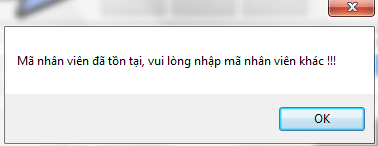
#### Trùng mã hàng hóa:



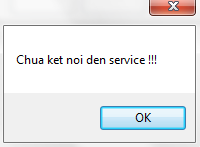
#### Trùng mã thành viên:



#### Trùng mã nhân viên:



### Lỗi chưa kết nối đến Service:



# KẾT LUẬN

* Ưu điểm:
  + Hoàn thành đề tài đúng thời gian đưa ra
  + Giao diện tương đối dễ sử dụng
  + Đáp ứng đầy đủ các chức năng cơ bản của việc quản lý một siêu thị
  + Một service nhưng có thể áp dụng cho nhiều đối tượng dùng
* Khuyết điểm:
  + Chưa xử lý được ưu đãi cho khách hàng thành viên
  + Chưa có hiệu chỉnh nhiều trong file app.config
  + Chỉ quản lý ở mức độ đơn giản, chưa quản lý ở mức cao hơn như thống kê, in ấn…
  + Chưa xử lý việc import dữ liệu từ các file word, excel, txt…
  + Chưa hiện thực chức năng mua hàng online.
* Ứng dụng vào thực tế:
  + Có thể áp dụng service cho những cửa hàng nhỏ
* Hướng mở rộng:
  + Tích hợp lên website để mua hàng online
  + Bổ sung phương thức tìm kiếm thông minh
  + Hoàn chỉnh ứng dụng giành cho khách hàng.

# TÀI LIỆU THAM KHẢO

* <http://www.msdn.microsoft.com>
* <http://www.congdongcviet.com>
* <http://www.codeproject.com>