Bài thực hành 02:

THỰC HÀNH XÂY DỰNG BIỂU ĐỒ LỚP, BIỂU ĐỒ TRẠNG THÁI

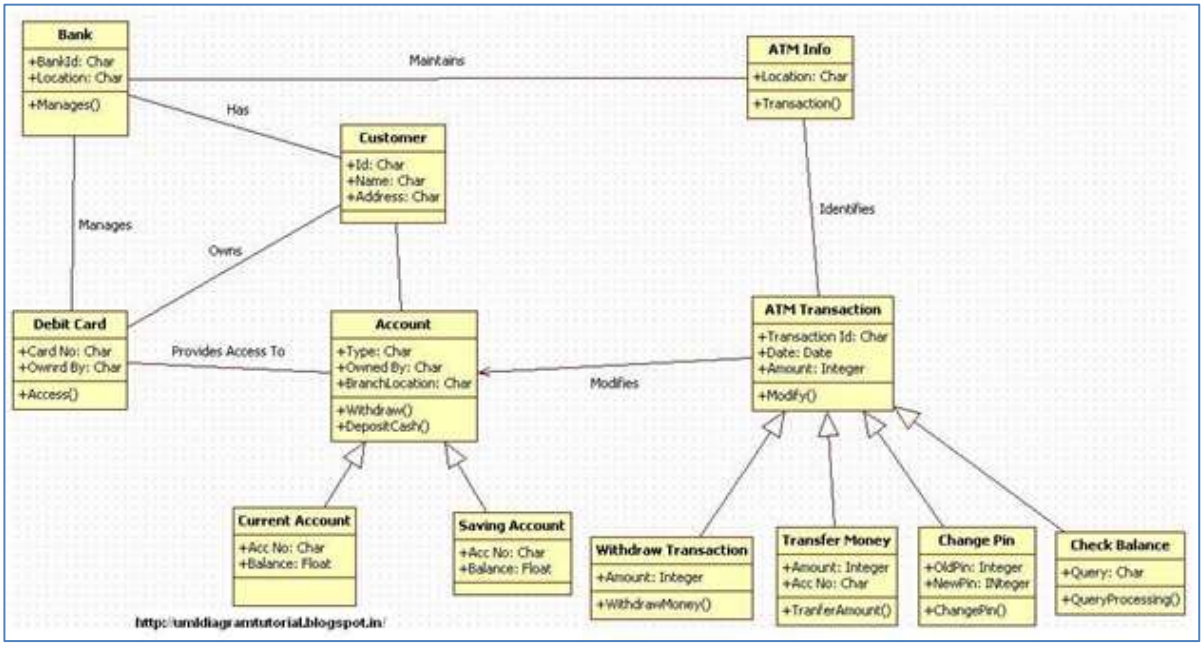
1. Mục tiêu:

- ✓ Trình bày được các thành phần của biểu đồ lớp, biểu đồ trạng thái
- ✓ Xác định được các lớp cơ bản, các phương thức và thuộc tính của các lớp cơ bản đó
- ✓ Sử dụng thành thạo phần mềm để biểu diễn biểu đồ lớp của hệ thống
- ✓ Xây dựng được biểu đồ lớp thực thể
- ✓ Xây dựng được biểu đồ lớp phân tích cho từng Use case dựa vào kịch bản Use case.
- ✓ Xây dựng được biểu đồ trạng thái của một lớp trong ứng dụng, trong một Use case.

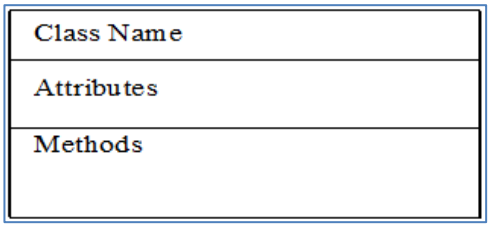
2. Cơ sở lý thuyết:

2.1. Các thành phần trong bản vẽ Class

Trước tiên, chúng ta xem một bản vẽ Class:

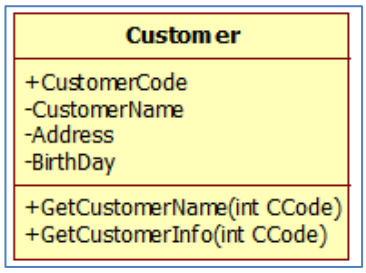


Classes (Các lớp): Class là thành phần chính của bản vẽ Class Diagram. Class mô tả về một nhóm đối tượng có cùng tính chất, hành động trong hệ thống. Ví dụ mô tả về khách hàng "Customer". Class được mô tả gồm tên Class, thuộc tính và phương thức.



Trong đó:

- ✓ Class Name: là tên của lớp.
- ✓ Attributes (thuộc tính): mô tả tính chất của các đối tượng. Ví dụ như khách hàng có Mã khách hàng, Tên khách hàng, Địa chỉ, Ngày sinh v.v…
- ✓ Method (Phương thức): chỉ các hành động mà đối tượng này có thể thực hiện trong hệ thống. Nó thể hiện hành vi của các đối tượng do lớp này tạo ra.

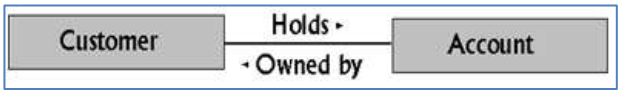


Relationship (Quan hệ): Relationship thể hiện mối quan hệ giữa các Class với nhau. các quan hệ thường sử dụng như sau:

- ✓ Association
- ✓ Aggregation
- ✓ Composition
- ✓ Generalization

+ *Association:*

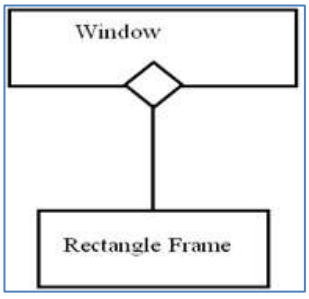
Association là quan hệ giữa hai lớp với nhau, thể hiện chúng có liên quan với nhau. Association thể hiện qua các quan hệ như "has: có", "Own: sở hữu" v.v…



Ví dụ quan hệ trên thể hiện Khách hàng nắm giữ Tài khoản và Tài khoản được sở hữu bởi Khách hàng.

+ *Aggregation:*

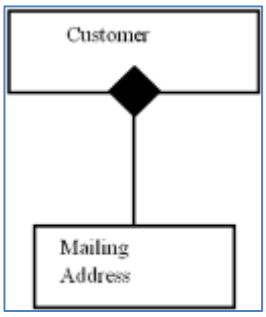
Aggregation là một loại của quan hệ Association nhưng mạnh hơn. Nó có thể cùng thời gian sống (cùng sinh ra hoặc cùng chết đi)



Ví dụ quan hệ trên thể hiện lớp Window(cửa sổ) được lắp trên Khung cửa hình chữ nhật. Nó có thể cùng sinh ra cùng lúc.

+ Composition

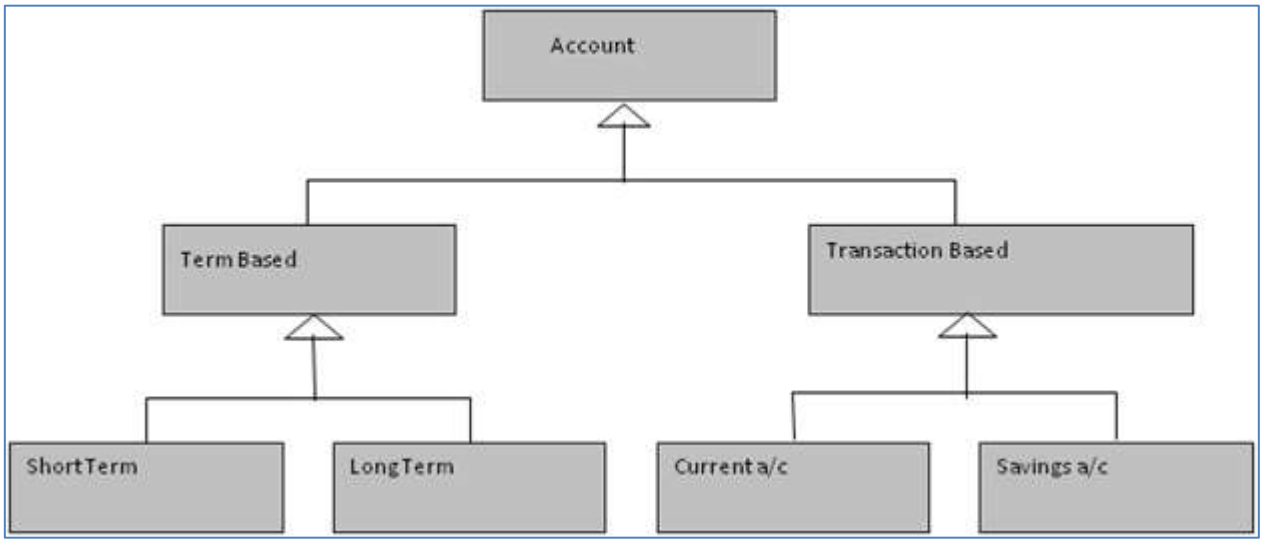
Composition là một loại mạnh hơn của Aggregation thể hiện quan hệ class này là một phần của class kia nên dẫn đến cùng tạo ra hoặc cùng chết đi.



Ví dụ trên class Mailing Address là một phần của class Customer nên chỉ khi nào có đối tượng Customer thì mới phát sinh đối tượng Mailing Address.

+Generalization

Generalization là quan hệ thừa kế được sử dụng rộng rãi trong lập trình hướng đối tượng.



Các lớp ở cuối cùng như Short Term, Long Term, Curent a/c, Savings a/c gọi là các lớp cụ thể (concrete Class). Chúng có thể tạo ra đối tượng và các đối tượng này thừa kế toàn bộ các thuộc tính, phương thức của các lớp trên.

Các lớp trên như Account, Term Based, Transaction Based là những lớp trừu tượng (Abstract Class), những lớp này không tạo ra đối tượng.

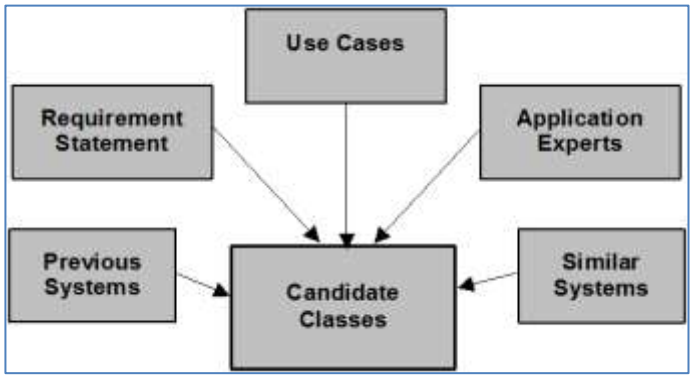
2.2. Cách xây dựng bản vẽ Class

Class Diagram là bản vẽ khó xây dựng nhất so với các bản vẽ khác trong OOAD và UML. Bạn phải hiểu được hệ thống một cách rõ ràng và có kinh nghiệm về lập trình hướng đối tượng mới có thể xây dựng thành công bản vẽ này.

Thực hiện theo các bước sau đây để xây dựng Class Diagram.

Bước 1: Tìm các Classes dự kiến

Entity Classes(các lớp thực thể) là các thực thể có thật và hoạt động trong hệ thống, bạn dựa vào các nguồn sau để xác định chúng.



- ✓ **Requirement statement**: Các yêu cầu. Chúng ta phân tích các danh từ trong các yêu cầu để tìm ra các thực thể.
- ✓ **Use Cases**: Phân tích các Use Case sẽ cung cấp thêm các Classes dự kiến.
- ✓ **Previous và Similar System:** có thể sẽ cung cấp thêm cho bạn các lớp dự kiến.
- ✓ **Application Experts:** các chuyên gia ứng dụng cũng có thể giúp bạn.

Xem xét, ví dụ ATM ở trên chúng ta có thể thấy các đối tượng là Entity Class như sau:

- ✓ **Customers**: khách hàng giao dịch là một thực thể có thật và quản lý trong hệ thống.
- ✓ **Accounts:** Tài khoản của khách hàng cũng là một đối tượng thực tế.
- ✓ **ATM Cards**: Thẻ dùng để truy cập ATM cũng được quản lý trong hệ thống.
- ✓ **ATM Transactions**: Các giao dịch được lưu giữ lại, cũng là một đối tượng có thật.
- ✓ **Banks**: Thông tin ngân hàng bạn đang giao dịch, nếu có nhiều nhà Bank tham gia vào hệ thống bạn phải quản lý nó. Lúc đó Bank trở thành đối tượng bạn phải quản lý.
- ✓ **ATM**: Thông tin ATM bạn sẽ giao dịch. Nó cũng được quản lý tương tự như Banks.

Lưu ý: Chỉ các thực thể bên trong hệ thống được xem xét, các thực thể bên ngoài hệ thống không được xem xét. Ví dụ Customers là những người khách hàng được quản lý trong hệ thống chứ không phải người dùng máy ATM bên ngoài. Bạn phải lưu ý điều này để phân biệt Class và Actor.

Bước 2: Tìm các thuộc tính và phương thức cho lớp

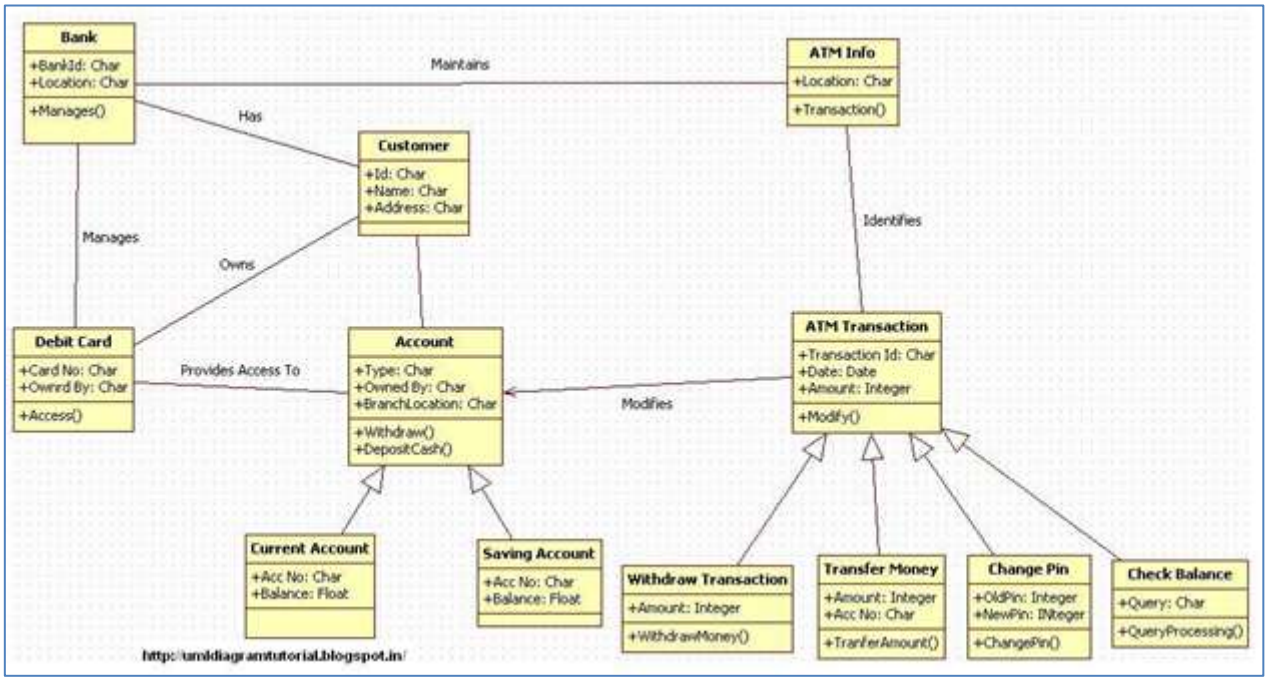
- ✓ **Tìm thuộc tính**: phân tích thông tin từ các form mẫu có sẵn, bạn sẽ tìm ra thuộc tính cho các đối tượng của lớp. Ví dụ các thuộc tính của lớp Customer sẽ thể hiện trên Form đăng ký thông tin khách hàng.
- ✓ **Tìm phương thức**: phương thức là các hoạt động mà các đối tượng của lớp này có thể thực hiện. Chúng ta sẽ bổ sung phương thức đầy đủ cho các lớp khi phân tích Sequence Diagram sau này.

Bước 3: Xây dựng các quan hệ giữa các lớp và phát hiện các lớp phát sinh

Phân tích các quan hệ giữa các lớp và định nghĩa các lớp phát sinh do các quan hệ sinh ra. Chúng ta phân tích các thực thể ở trên và nhận thấy.

- Lớp ***Accounts*** có thể chia thành nhiều loại tài khoản như ***Current Accounts*** và ***Saving Accounts*** và có quan hệ thừa kế với nhau.
- Lớp ***ATM Transactions*** cũng có thể chia thành nhiều loại giao dịch như ***Deposit, Withdraw, Transfer*** v.v.. và chúng cũng có quan hệ thừa kế với nhau.

- ✓ Tách và vẽ chúng lên bản vẽ, sẽ có Class Diagram cho hệ thống ATM như sau:



2.3. Đặc tả Class

Nhìn vào Class Diagram chúng ta có thể thấy cấu trúc của hệ thống gồm những lớp nào nhưng để cài đặt chúng, chúng ta phải đặc tả chi tiết hơn nữa. Trong đó, cần mô tả:

- ✓ Các thuộc tính: Tên, kiểu dữ liệu, kích thước
- ✓ Các phương thức:

+ Tên

+ Mô tả

+ Tham số đầu vào: Tên, kiểu dữ liệu, kích thươcs

+ Kết quả đầu ra: Tên, kiểu dữ liệu, kích thước

+ Luồng xử lý

+ Điều kiện bắt đầu

+ Điều kiện kết thúc

2.4. Sử dụng bản vẽ Class

Có thể tóm tắt một số ứng dụng của bản vẽ Class Diagram như sau:

- ✓ Hiểu cấu trúc của hệ thống
- ✓ Thiết kế hệ thống
- ✓ Sử dụng để phân tích chi tiết các chức năng (Sequence Diagram, State Diagram v.v…)
- ✓ Sử dụng để cài đặt (coding)

Bài tập 1:

"Một công ty chuyên kinh doanh về các thiết bị điện tử và công nghệ thông tin trong nhiều năm nay và đã có một lượng khách hàng nhất định.Để mở rộng hoạt động kinh doanh của mình, công ty mong muốn xây dựng một hệ thống thương mại điện tử nhằm mở rộng phạm vi kinh doanh trên mạng Internet.

Hệ thống mới phải đảm bảo cho khách hàng viếng thăm Website dễ dàng lựa chọn các sản phẩm, xem các khuyến mãi cũng như mua hàng. Việc thanh toán có thể được thực hiện qua mạng hoặc thanh toán trực tiếp tại cửa hàng. Khách hàng có thể nhận hàng tại cửa hàng hoặc sử dụng dịch vụ chuyển hàng có phí của công ty.

Ngoài ra, hệ thống cũng cần có phân hệ để đảm bảo cho công ty quản lý các hoạt động kinh doanh như số lượng hàng có trong kho, quản lý đơn đặt hàng, tình trạng giao hàng, thanh toán v.v…

Thông tin chi tiết các chức năng các bạn có thể tham khảo thêm tại các Website bán hàng.

1. Xây dựng Class Diagram cho hệ thống eCommerce

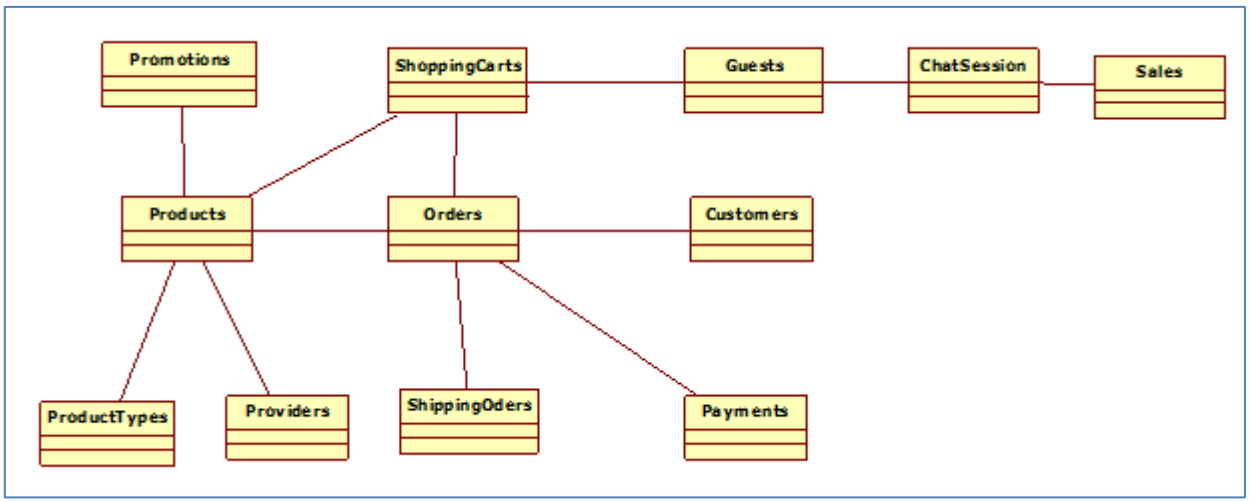
Bước 1: Tìm các Classes dự kiến

Nghiên cứu kỹ các yêu cầu, Use Case và nghiên cứu kỹ các hệ thống tương tự để xác định các lớp dự kiến thông qua việc xác định các đối tượng có trong hệ thống.

Xem xét Use Case Diagram của hệ thống:

- ✓ Phân tích Use Case "**Xem sản phẩm**" chúng ta xác định thực thể sản phẩm (**Products**). Sản phẩm được phân loại theo chủng loại (**Product Types**) và Nhà sản xuất (**Providers**) nên đây có thể là 2 lớp có quan hệ với class **Products.**
- ✓ Xem xét Use Case "**Xem khuyến mãi**" xác định Class Chương trình khuyến mãi (**Promotions**)
- ✓ Use Case "**Quản lý giỏ hàng**" -> Class giỏ hàng (**Shopping Carts**)
- ✓ Use Case Chat -> Class **Chat session**. Những người dùng tham gia Chat là **Sales** và **Guest** có thể là hai class dự kiến.
- ✓ Use Case "**Đăng ký thành viên**" -> Khách hàng (**Customers**)
- ✓ Use Case "**Quản lý đơn hàng"** -> Class đơn hàng (**Orders**), class thu tiền (**Payments**) và Quản lý chuyển hàng (**Shipping Orders**) có thể là 2 lớp có liên quan với Class **Orders.**

Tạm thời vẽ và xác định quan hệ sơ bộ chúng ta có bản vẽ Class dự kiến như sau:



Bản vẽ này giúp chúng ta có cái nhìn cơ bản về cấu trúc hệ thống để tiếp tục phân tích. Tất nhiên, phân tích tất cả các Use Case còn lại và tìm hiểu thêm về hệ thống để bổ sung đầy đủ Class dự kiến cho hệ thống.

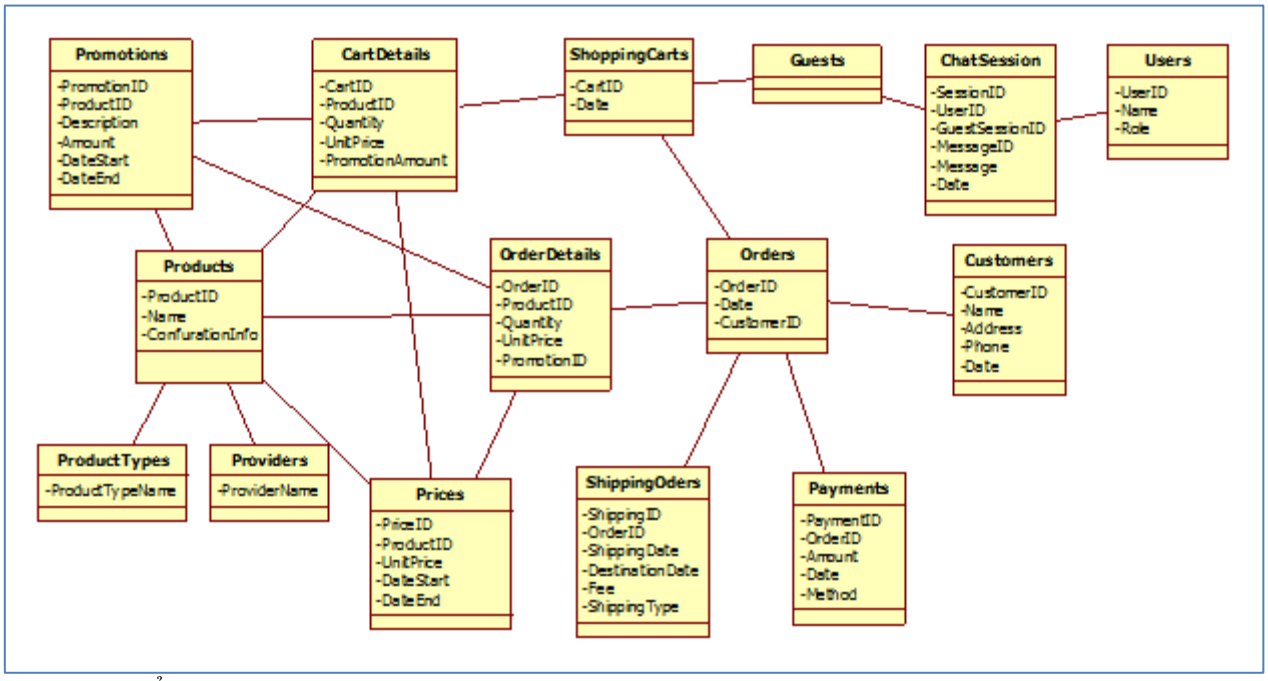
Bước 2: Xác định thuộc tính và quan hệ cho các lớp

Chúng ta bổ sung các thuộc tính cho các lớp và phân tích quan hệ của chúng.

- ✓ **Products**: xem xét tài liệu mô tả sản phẩm của hệ thống chúng ta có thể thấy Class **Products** cần những thuộc tính sau: Tên sản phẩm, mô tả, cấu hình, Giá bán, khuyến mãi, bảo hành (xem mô tả chi tiết sản phẩm trên Website)… Trong đó, thuộc tính giá thay đổi theo thời gian nên chúng ta nên tách ra thành lớp riêng là Giá (**Prices**). Tương tự thuộc tính khuyến mãi cũng được tách ra thành lớp **Promotions.**
- ✓ **Prices**: có các thuộc tính là Mã sản phẩm, Giá, ngày bắt đầu, ngày hết hạn.

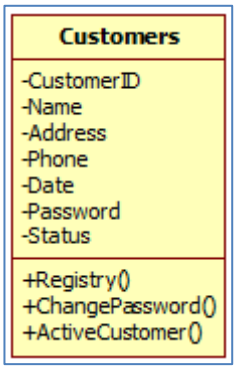
- ✓ **Promotions**: tương tự như giá nó cần có lớp riêng với các thuộc tính là Mã sản phẩm, Mô tả khuyến mãi, Giá trị khuyến mãi, Ngày bắt đầu, Ngày hết hạn.
- ✓ **ProductTypes**: chứa loại sản phẩm
- ✓ **Providers**: chưa tên nhà sản xuất
- ✓ **ShoppingCarts**: chứa các thông tin như: cartID, ngày, mã sản phẩm, số lượng, đơn giá. Chúng ta nhận thấy nếu để nguyên lớp này khi tạo đối tượng chúng sẽ lặp thông tin cartID và ngày mua nên tách chúng ra thành **ShoppingCarts** với các thuộc tính CartID, ngày và **CartDetails** với các thuộc tính ProductID, số lượng, đơn giá.
- ✓ Tương tự chúng ta có class **Orders** với OrderID, ngày, customerID và class **Orderdetails** với ProductID, số lượng, đơn giá.
- ✓ **Payments**: chứa các thông tin như PaymentID, OrderID, ngày trả, số tiền, hình thức thanh toán.
- ✓ **Shippings**: có thể chứa ShippingID, OrderID, Ngày chuyển, ngày đến, số tiền, phương thức vận chuyển.
- ✓ **Customers**: CustomerID, Họ và tên, địa chỉ, điện thoại, ngày đăng ký v.v…
- ✓ **Guests**: có thể chứa sessionID để xác định thông tin khi chat
- ✓ **Sales**: có thể gộp với lớp người dùng (**Users**) chứa UserID, Name
- ✓ **ChatSessions**: ChatsessionID, tên người bán hàng, mã khách, mã tin nhắn, nội dung tin nhắn, ngày.

Nhập đầy đủ thuộc tính và vẽ, chúng ta có bản vẽ như sau:



Bước 3: Bổ sung phương thức cho các lớp

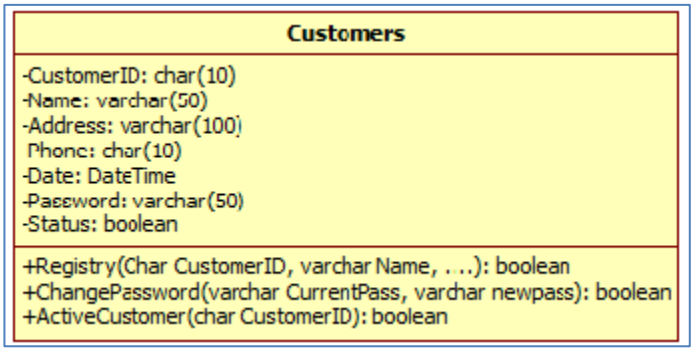
Phương thức là các hành động mà đối tượng sinh ra từ lớp đó có thể thực hiện trong hệ thống. Ví dụ các đối tượng của lớp **Customer**s có thể đăng ký mới, có thể thay đổi mật khẩu (password), kích hoạt người dùng (Active) v.v..



Bước 4: Thiết kế chi tiết các thuộc tính và phương thức cho lớp

Khi đã có được Class Diagram, bạn cần thiết kế chi tiết các lớp bằng cách đặc tả các thuộc tính và phương thức của nó.

- ✓ Đặc tả thuộc tính: chúng ta xác định kiểu dữ liệu và kích thước.
- ✓ Đặc tả phương thức: chúng ta xác định dữ liệu đầu vào, dữ liệu đầu ra.



Việc sử dụng các kiểu dữ liệu và mô tả các phương thức của một lớp chúng ta đã học trong lập trình hướng đối tượng.

***Bài tập 2:* Mô tả chức năng và yêu cầu của HỆ THỐNG QUẢN LÝ THƯ VIỆN**

Xây dựng hệ thống quản lý thư viện cho trường Đại học gồm các hoạt động quản lý thông tin sách, quản lý thông tin độc giả, quản lý hoạt động mượn trả sách:

- ✓ Sinh Viên của trường muốn mượn sách của thư viện thì trước tiên phải đăng ký làm thẻ thư viện theo lớp, thông tin về thẻ thư viện gồm (Mã độc giả, họ tên, lớp, ngày sinh, giới tính), khi đó thủ thư thực hiện nhập thông tin về thẻ thư viện vào hệ thống và in thẻ thư viện giao cho sinh viên, khi thông tin về thẻ thư viện có sai sót hệ thống cho phép thủ thư sửa, khi độc giả bị loại bỏ khỏi thư viện hệ thống cho phép xóa thẻ thư viện.

- ✓ Các cuốn sách trong thư viện được quản lý thông tin theo đầu sách, mỗi đầu sách trong thư viện có nhiều bản sao khác nhau. Thông tin về đầu sách gồm (Mã đầu sách, tên đầu sách, nhà xuất bản, số trang, kích thước, tác giả, số lượng sách), thông tin về bản sao các đầu sách gồm (mã đầu, mã sách, tình trạng, ngaynhap). Khi thư viện nhập sách mới về thủ thư có nhiệm vụ nhập thông tin sách vào trong thư viện, nếu thông tin về sách có thay đổi hoặc loại bỏ ra khỏi thư viện, thủ thư thực hiện sửa thông tin sách hoặc xóa sách.
- ✓ Thư viện quản lý các đầu sách theo các chuyên ngành, các đầu sách được phân thành các chuyên ngành khác nhau. Thông tin chuyên ngành gồm (Mã chuyên ngành, tên chuyên ngành, mô tả).
- ✓ Mỗi một độc giả một lần mượn chỉ được mượn một cuốn sách, khi độc giả muốn mượn sách vào tìm sách trong thư viện và ghi thông tin vào phiếu mượn gồm mã sách, mã độc giả và gửi cho thủ thư. Thủ thư tiến hành ghi nhận thông tin phiếu mượn vào trong hệ thống, giữ lại thẻ của độc giả và giao sách cho độc giả. Thông tin phiếu mượn gồm (Mã sách, mã độc giả, mã thủ thư cho mượn sách, ngày mượn, tình trạng).
- ✓ Khi độc giả trả sách thủ thư thực hiện chức năng trả sách để ghi nhận tình trạng trả sách cho phiếu mượn.
- ✓ Định kỳ thủ thư phải làm các báo cáo thống kê gửi lên lãnh đạo thư viện các báo cáo gồm: Thông tin các đầu sách cho mượn nhiều nhất, thông tin về các độc giả chưa trả sách.
- ✓ Để quản lý người dùng hệ thống, trong thư viện có một nhân viên đóng người quản trị vai trò làm. Nhân viên này có quyền quản lý thông tin người dùng hệ thống. Khi có nhân viên thư viện mới người quản trị cập nhật thông tin thủ thư vào hệ thống, tạo tài khoản và cấp quyền cho nhân viên thư viện. Khi thông tin nhân viên thư viện có sai sót hoặc loại bỏ ra khỏi hệ thống thì người quản trị sửa hoặc xóa thông tin nhân viên thư viện ra khỏi hệ thống.
- ✓ Người dùng hệ thống phải đăng nhập trước khi thực hiện.

Yêu cầu:

1. Dựa vào bản đặc tả trên, hãy xác định các lớp thực thể gồm (tên lớp, các thuộc tính cơ bản, các phương thức cơ bản).
2. Xác định mối quan hệ giữa các lớp, sử dụng Rational Rose để biểu diễn các lớp đó.

3. Phân tích một Use case dựa vào kịch bản, ví dụ Use case thêm đầu sách mới xác định các lớp trong Use case đó.
4. Xây dựng được biểu đồ trạng thái của lớp DocGia trong ứng dụng dựa vào đặc tả phần mềm
5. Xây dựng biểu đồ trạng thái lớp DocGia trong Use case trả sách.
6. Phân tích các Use case còn lại trong hệ thống xây dựng biểu đồ lớp phân tích cho các Use case đó.
7. Xem xét các lớp còn lại, xây dựng biểu đồ chuyển trạng thái nếu có.

Hướng dẫn thực hiện:

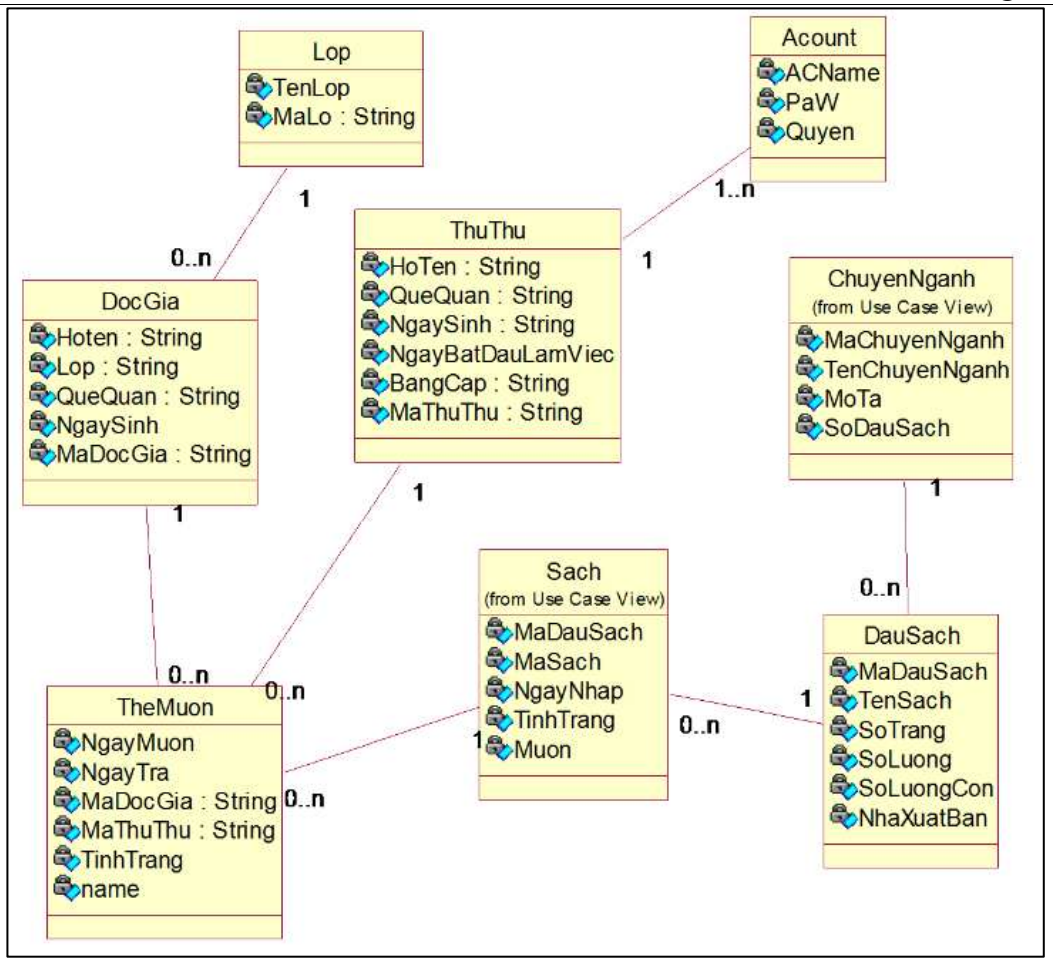
1. Dựa vào bản đặc tả trên, hãy xác định các lớp cơ bản bao gồm (tên lớp, các thuộc tính cơ bản, các phương thức cơ bản).

Phân tích bài toán:

- **Bước 1:** Xem xét kỹ lại đặc tả hệ thống, kịch bản Use case, sử dụng phương pháp trích chọn danh từ: Sinh viên, lớp, thẻ thư viện, Mã độc giả, họ tên, lớp, ngày sinh, giới tính, thủ thư, sách, đầu sách, bản sao, mã đầu, mã sách, tình trạng, chuyên ngành, Mã chuyên ngành, tên chuyên ngành, mô tả, độc giả, phiếu mượn, Mã sách, mã độc giả, mã thủ thư cho mượn sách, ngày mượn, tình trạng, tài khoản, người dùng
- **Bước 2:** Loại bỏ các danh từ trung nhau hoặc không có giá trị. Ví dụ danh từ Sinh viên, thẻ thư viện, bạn đọc, độc giả đều chỉ một đối tượng người tham gia mượn sách thư viện là độc giả.
- **Bước 3:** Xác định các thuộc tính cho từng lớp. Các danh từ có thể trở thành lớp, và các danh từ khác có thể trở thành thuộc tính của lớp. Ví dụ danh từ phiếu mượn thành lớp PhieuMuon các danh từ Mã sách, mã độc giả, mã thủ thư cho mượn sách, ngày mượn, tình trạng thành thuộc tính của lớp PhieuMuon.
- **Bước 4:** Xác định quan hệ của các lớp

2. Xác định liên kết giữa các lớp nếu có.

Minh họa:



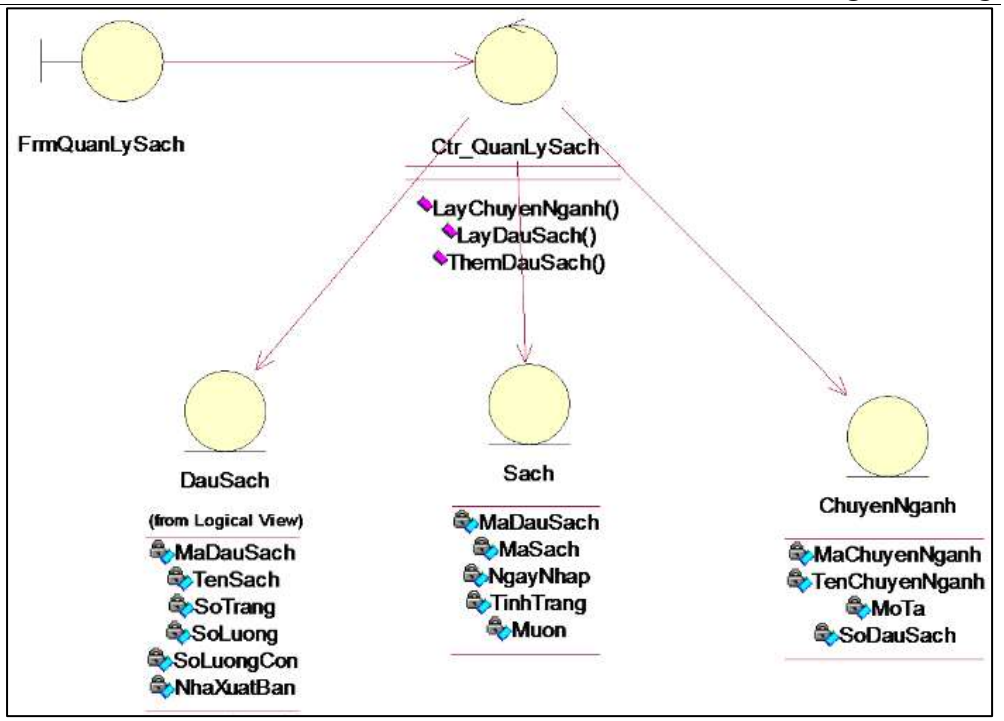
3. Sau khi xây dựng biểu đồ lớp dựa vào các danh từ, phân tích kịch bản cho từng Use case để xây dựng biểu đồ lớp cho từng Use case.

Hướng dẫn:

Phân tích: Dựa vào kịch bản trên người sử dụng phân tích Use case tìm ra các lớp.

- B1. Tìm lớp Bound, theo nguyên tắc giữa Actor và Use case có một lớp Bound.
- B2. Xác định lớp Control, theo nguyên tắc mỗi Use case có ít nhất một lớp Control.
- B3. Xác định các lớp thực thể, Đọc kịch bản xác định các danh từ tham gia Use case để tìm ra lớp thực thể.

Minh họa:

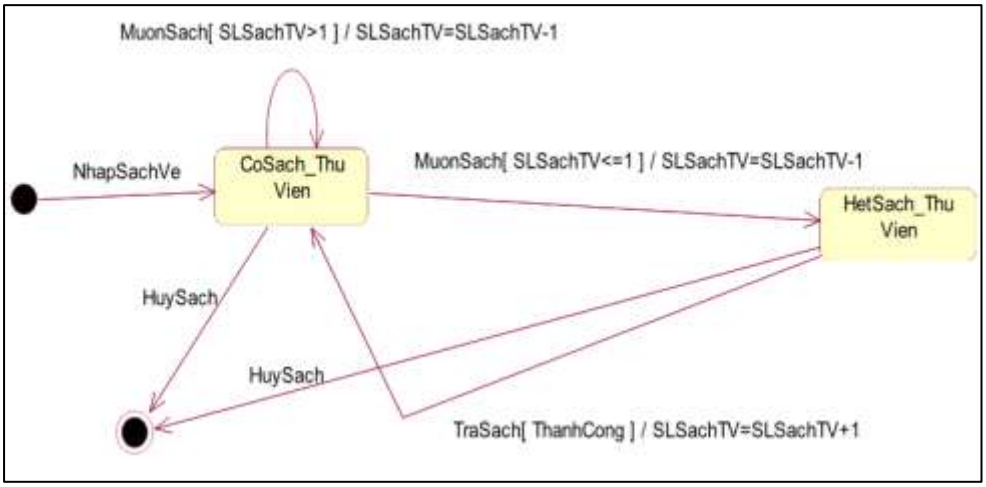


4. Xây dựng biểu đồ trạng thái

Hướng dẫn: Phân tích

- B1: Đọc đặc tả phần mềm, hoặc đặc tả chi tiết dựa vào nghiệp vụ hệ thống
- B2: Xác định các trạng thái có thể của lớp
- B2: Xác định các chuyển trạng thái, và sự kiện, điều kiện chuyển, hành động để chuyển giữa các trạng thái.

Minh họa:



Bài tập tự thực hiện:

1. Phân tích các Use case còn lại trong hệ thống xây dựng biểu đồ lớp phân tích cho các Use case đó.
2. Xem xét các lớp còn lại, xây dựng biểu đồ chuyển trạng thái nếu có.

---------- Hết Lab 02----------