

Chương 7: Thiết kế lớp

Thiết kế lớp đối tượng: Tinh chế thuộc tính

□ Kiểu thuộc tính: có 3 loại

- ⊙ Thuộc tính đơn trị
- ⊙ Thuộc tính đa trị: có thể dùng các cấu trúc list, array để khai báo cài đặt.
 - Ví dụ: thuộc tính *Số Điện Thoại* của lớp *Nhân Viên* có thể là đa trị
 - SốĐT[3]: String
 - SốĐT[1..3]: String
- ⊙ Thuộc tính dùng để tham chiếu tới các đối tượng khác hoặc tới một thể hiện kết nối

Thiết kế lớp đối tượng: Tinh chế thuộc tính

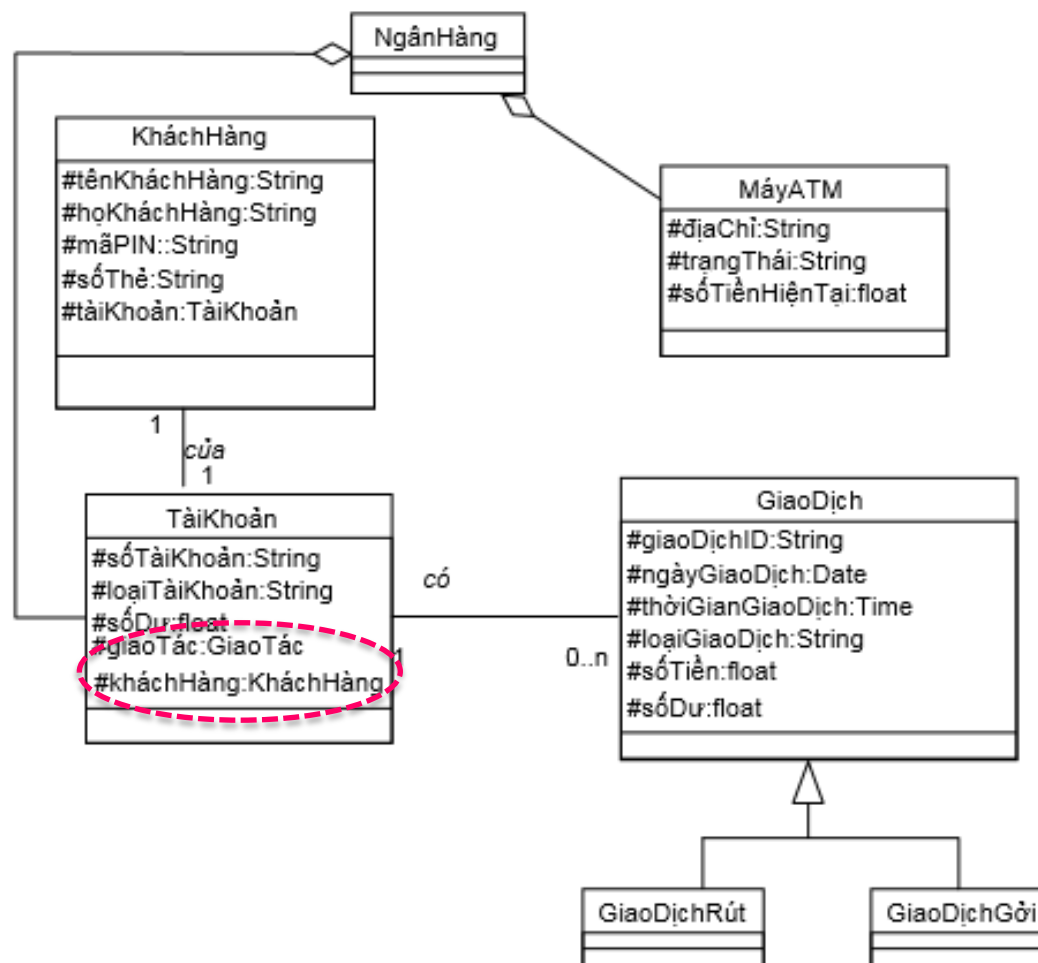
□ **Hiển thị thuộc tính**

- ⊙ *<Phạm vi> <tên> : <kiểu thuộc tính> = <giá trị khởi tạo>*
- ⊙ + : toàn cục
- ⊙ # : bảo vệ
- ⊙ - : cục bộ

Ghi chú: phạm vi mặc định của các thuộc tính là **cục bộ**

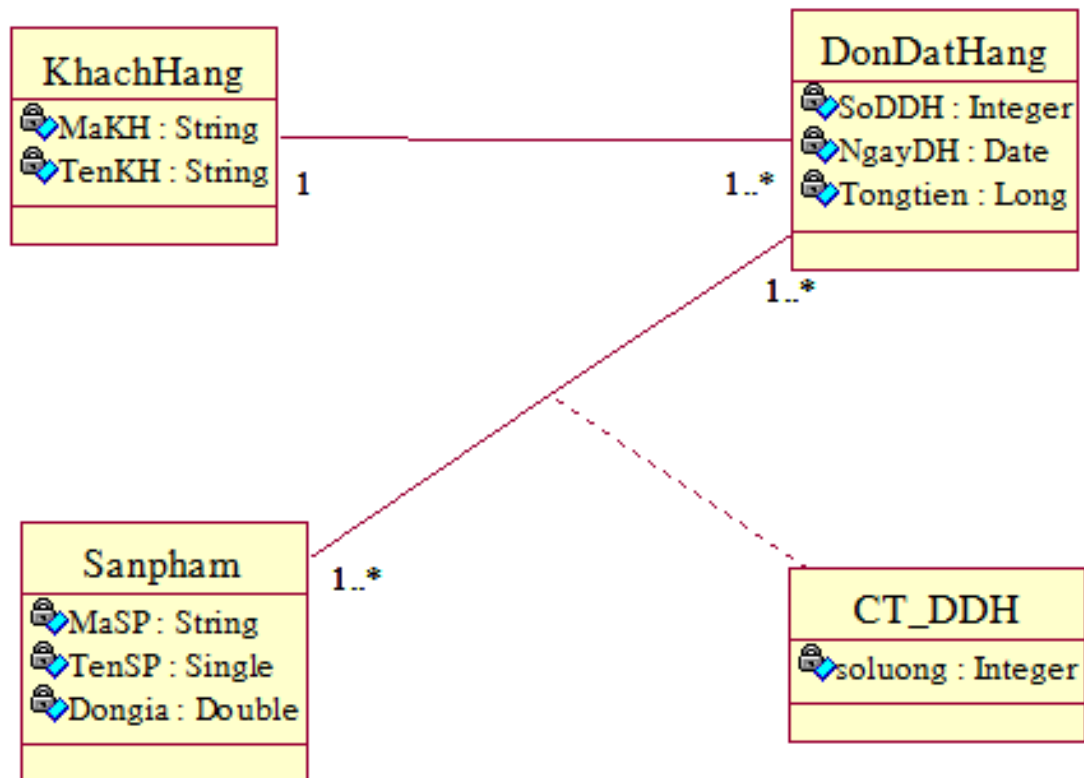
Thiết kế lớp đối tượng: Tinh chế thuộc tính

Ví dụ: sơ đồ mức phân tích



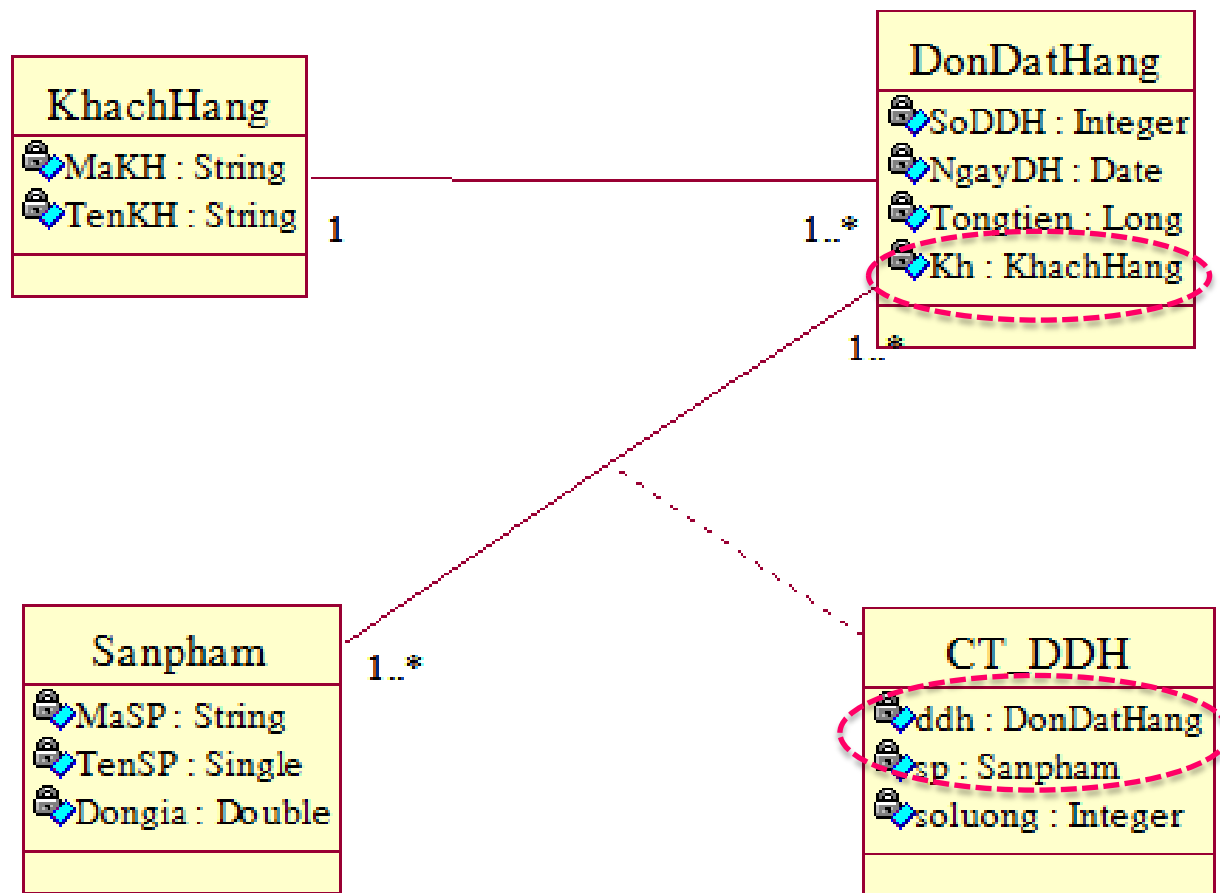
Thiết kế lớp đối tượng: Tinh chế thuộc tính

Ví dụ: sơ đồ mức phân tích



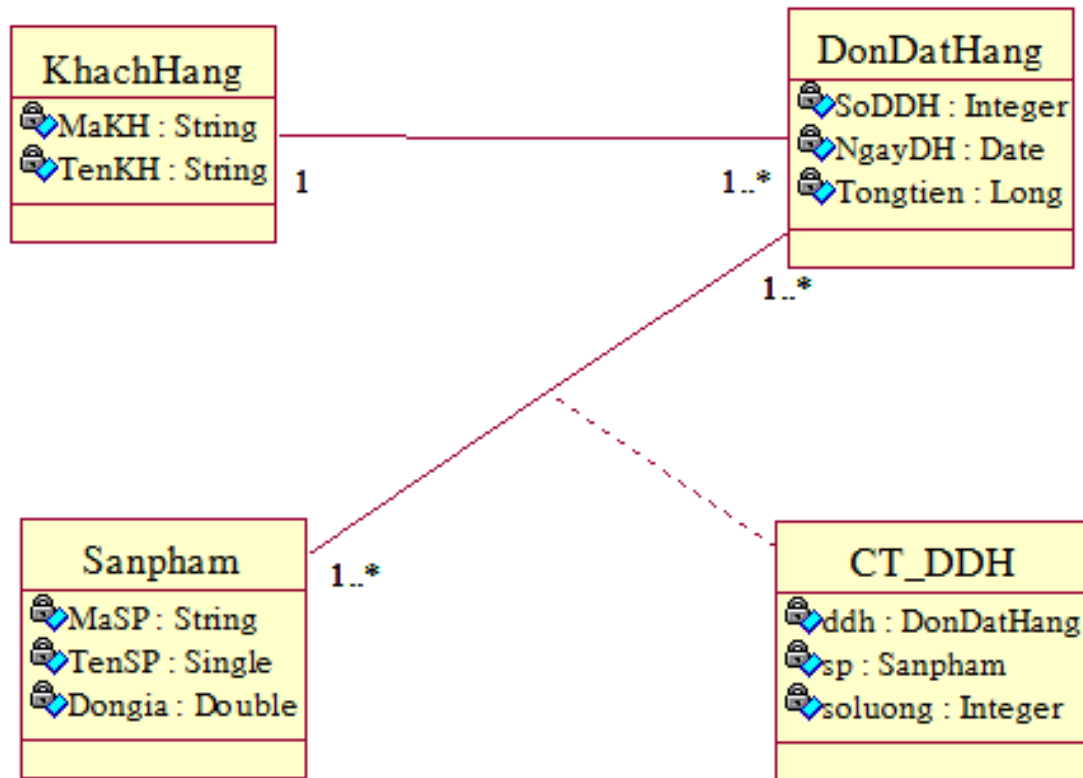
Thiết kế lớp đối tượng: Tính chế thuộc tính

Ví dụ: sơ đồ mức thiết kế sau khi tinh chỉnh thuộc tính



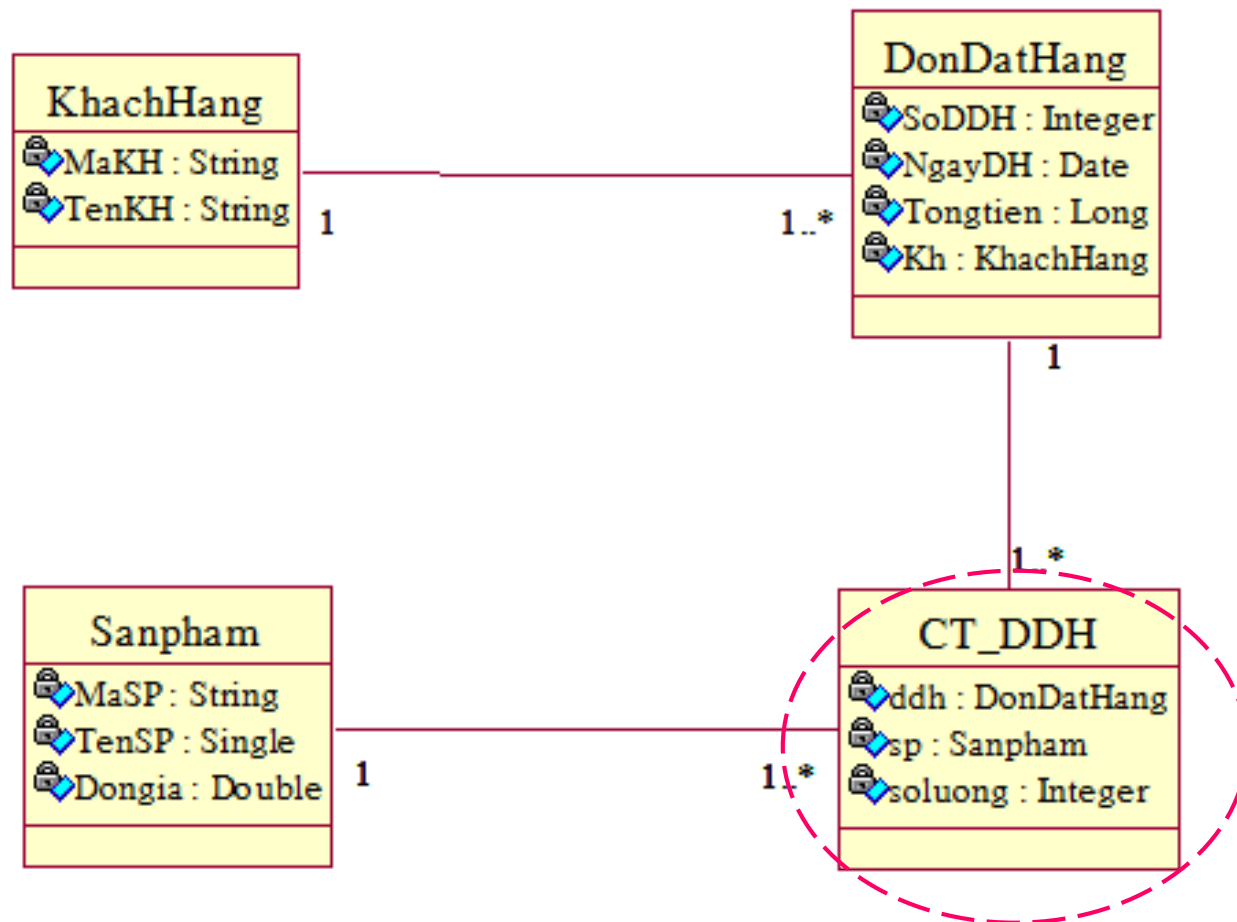
Thiết kế lớp đối tượng: Tinh chế mối kết hợp

□ Lớp kết hợp



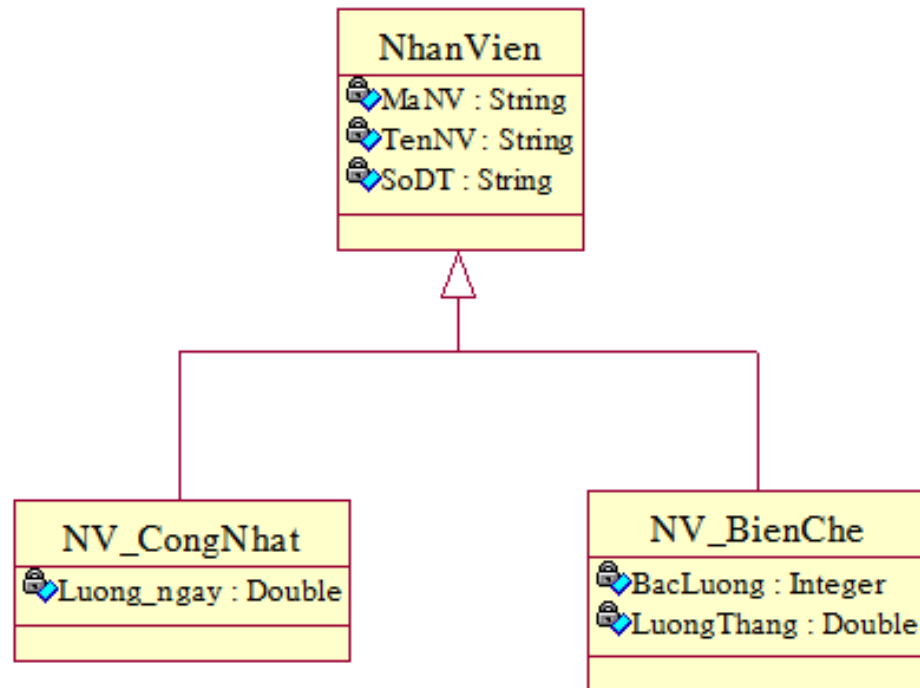
Thiết kế lớp đối tượng: Tinh chế mối kết hợp

□ Lớp kết hợp



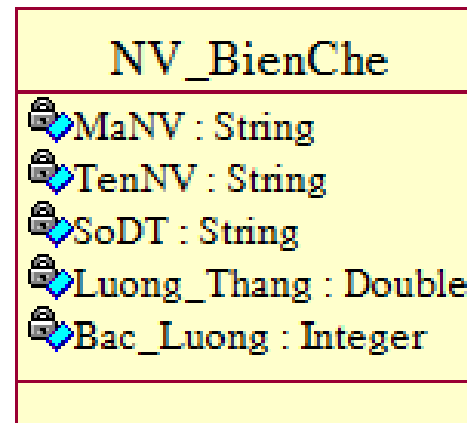
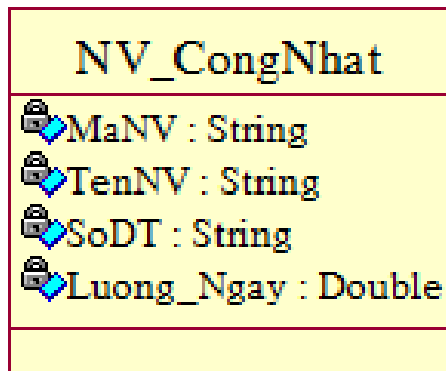
Thiết kế lớp đối tượng: Tinh chế mối kết hợp

- Tổng quát hóa và chuyên biệt hóa
Trường hợp 1:



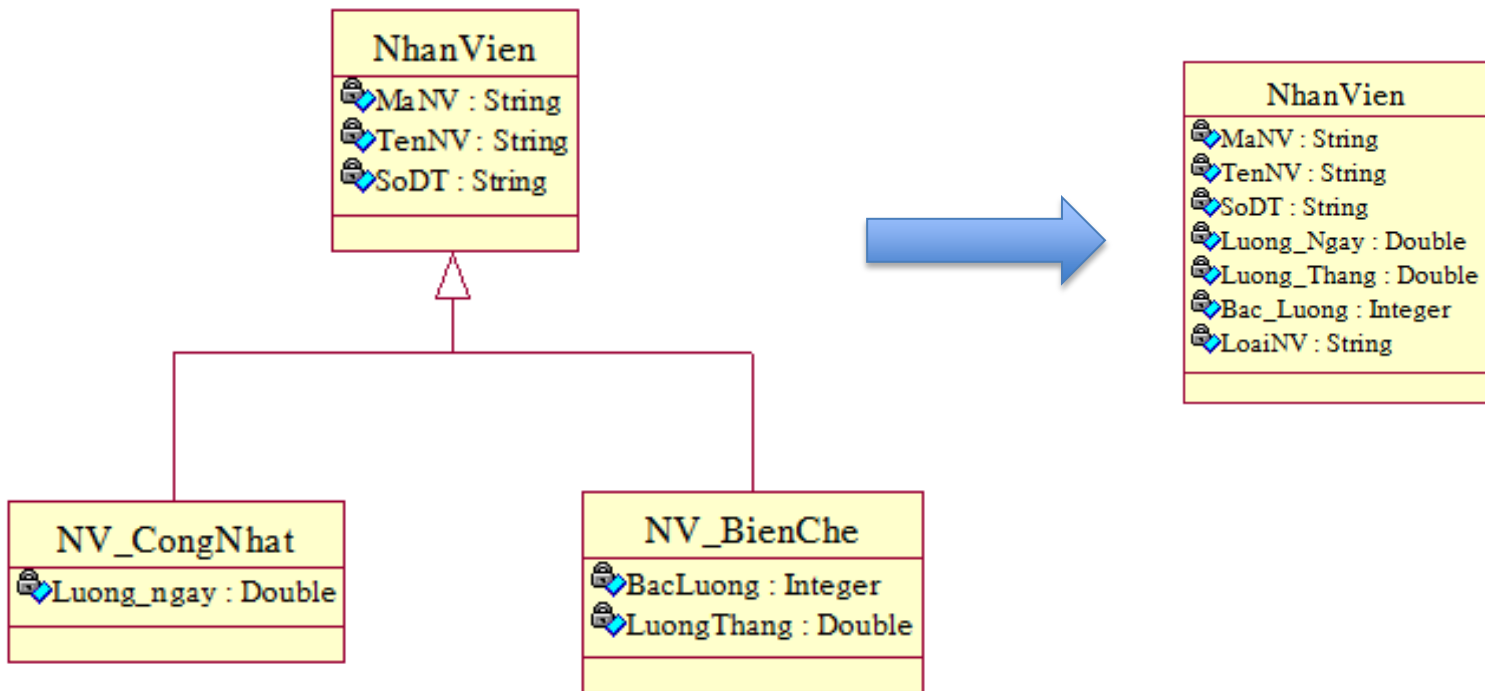
Thiết kế lớp đối tượng: Tinh chế mối kết hợp

- Tổng quát hóa và chuyên biệt hóa
Trường hợp 2:



Thiết kế lớp đối tượng: Tinh chế mối kết hợp

- Tổng quát hóa và chuyên biệt hóa
Trường hợp 3:



Thiết kế lớp đối tượng: Tinh chế phương thức

- Mục tiêu: Mô tả thuật toán cho các phương thức đã xác định được ở giai đoạn phân tích, bằng cách dùng mã giả hoặc sơ đồ.
- Áp dụng các hệ luật dẫn:
 - ⊙ Một tập lớn các lớp đơn giản sẽ tốt hơn một tập nhỏ các lớp phức tạp.
 - ⊙ Tạo một lớp tổng quát cho các lớp mà chúng ta thấy nội dung giống nhau, mục tiêu là tái sử dụng.
 - ⊙ Luôn tập trung vào mục tiêu của lớp khi định nghĩa, nhằm tránh thiết kế lạc đề hoặc mở rộng vượt khỏi phạm vi ý nghĩa lớp.

Thiết kế lớp đối tượng: Tinh chế phương thức

- Các phương thức để cài đặt một lớp có thể là:
 - ◉ *Constructor* : phương thức tạo thể hiện (đối tượng) của lớp
 - ◉ *Destructor*: phương thức hủy thể hiện của lớp
 - ◉ *Conversion*: phương thức chuyển đổi một đơn vị đo lường này sang một đơn vị đo lường khác
 - ◉ *Copy*: phương thức sao chép nội dung của một thể hiện sang một thể hiện khác
 - ◉ *Attribute set*: phương thức gán giá trị cho một hoặc nhiều thuộc tính
 - ◉ *Attribute get*: phương thức trả về giá trị của một hoặc nhiều thuộc tính
 - ◉ *I/O method*: phương thức cung cấp tới hoặc nhận dữ liệu từ một thiết bị
 - ◉ *Domain specific*: phương thức xác định tới các ứng dụng của đối tượng

Thiết kế lớp đối tượng: Tinh chế phương thức

□ Hiện thị phương thức

<phạm vi> <tên> <(danh sách tham số)>:<kiểu trả về>

⊙ Trong đó:

- *<phạm vi>* được qui định như thuộc tính.
- *<danh sách tham số>*: mỗi tham số cách nhau bởi dấu phẩy và có cú pháp: *<tên tham số>:<kiểu dữ liệu> = <giá trị mặc định>*
- *<kiểu trả về>*: là một đặc tả độc lập ngôn ngữ về cài đặt giá trị trả về của phương thức. Nếu mục này bị bỏ qua thì phương thức không trả về giá trị.

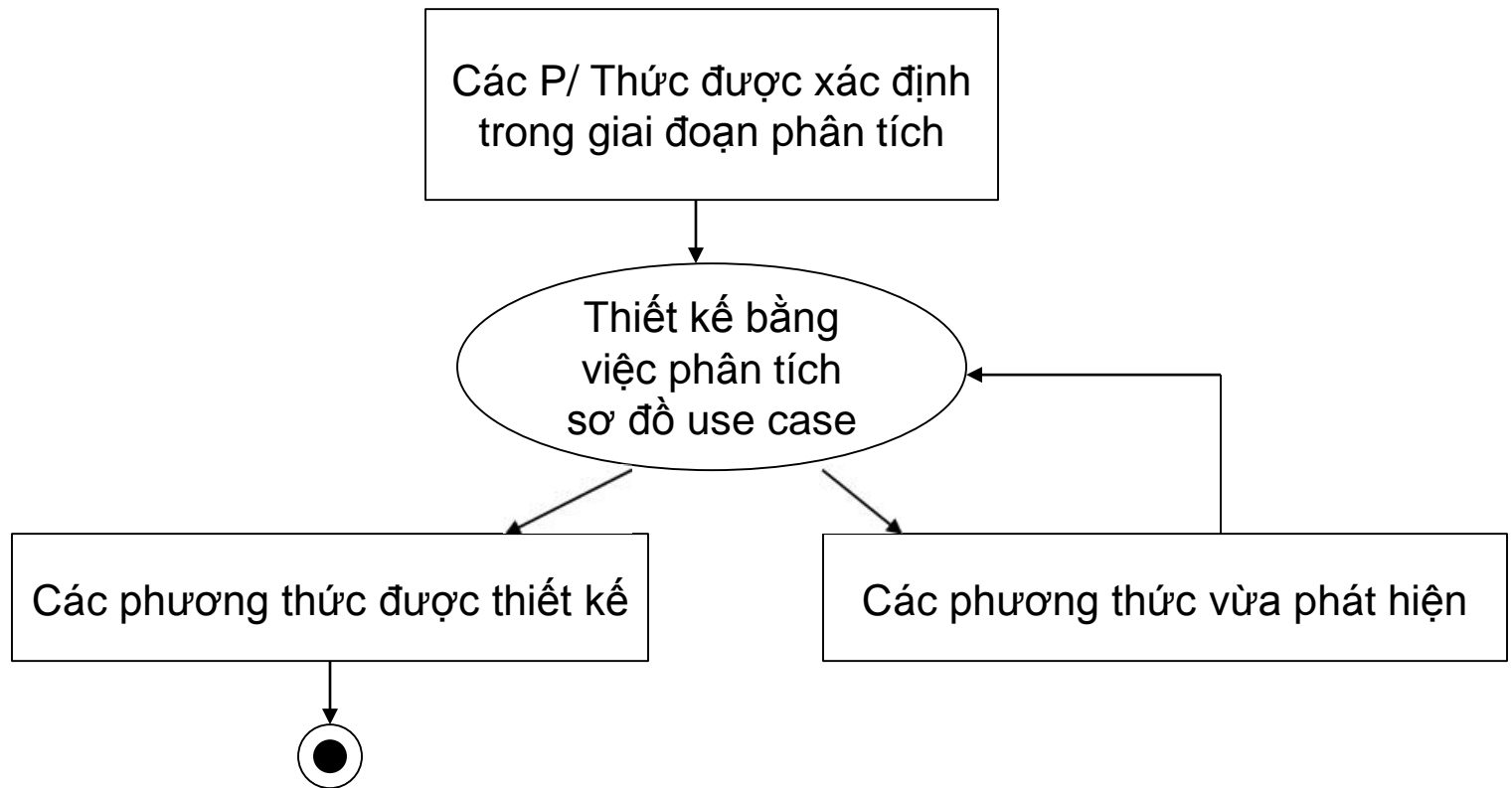
⊙ Các phương thức đa số là các phương thức có phạm vi toàn cục.

⊙ Ví dụ:

- `+get_Tên() : String`
- `+get_SốTàiKhoản(vtàiKhoản : TàiKhoản) : String`

Thiết kế lớp đối tượng: Tinh chế phương thức

◉ Thiết kế nội dung phương thức

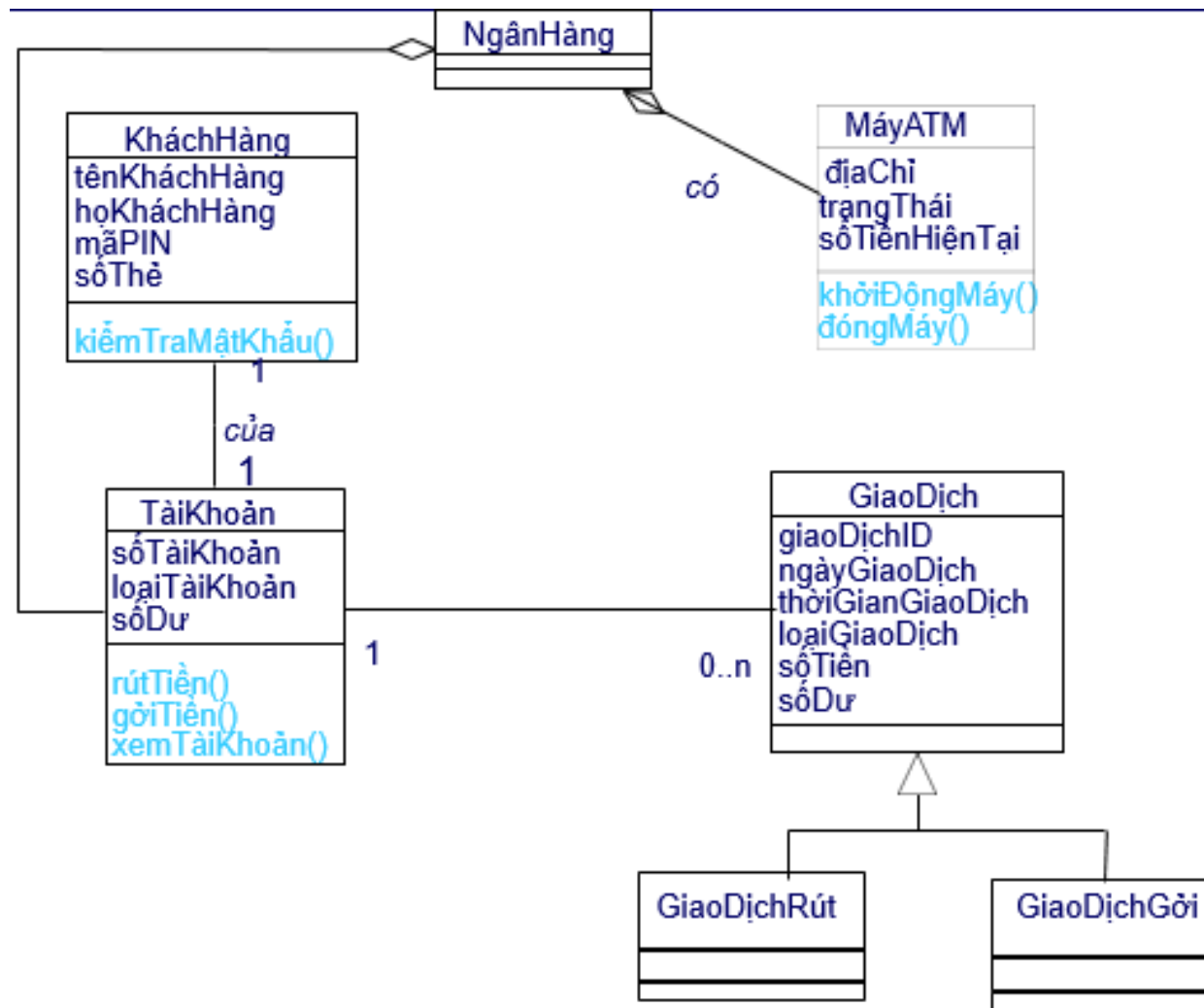


Thiết kế lớp đối tượng: Tinh chế phương thức

□ Ví dụ: Các phương thức hệ thống ATM trong giai đoạn phân tích

- `KháchHàng:: kiểmTraMậtKhẩu()`
- `TàiKhoản:: rútTiền()`
- `TàiKhoản:: gửiTiền()`
- `TàiKhoản:: xemThôngTin()`
- `MáyATM:: khởiĐộngMáy()`
- `MáyATM:: đóngMáy()`

Thiết kế lớp đối tượng: Tinh chế phương thức



Thiết kế lớp đối tượng: Tinh chế phương thức

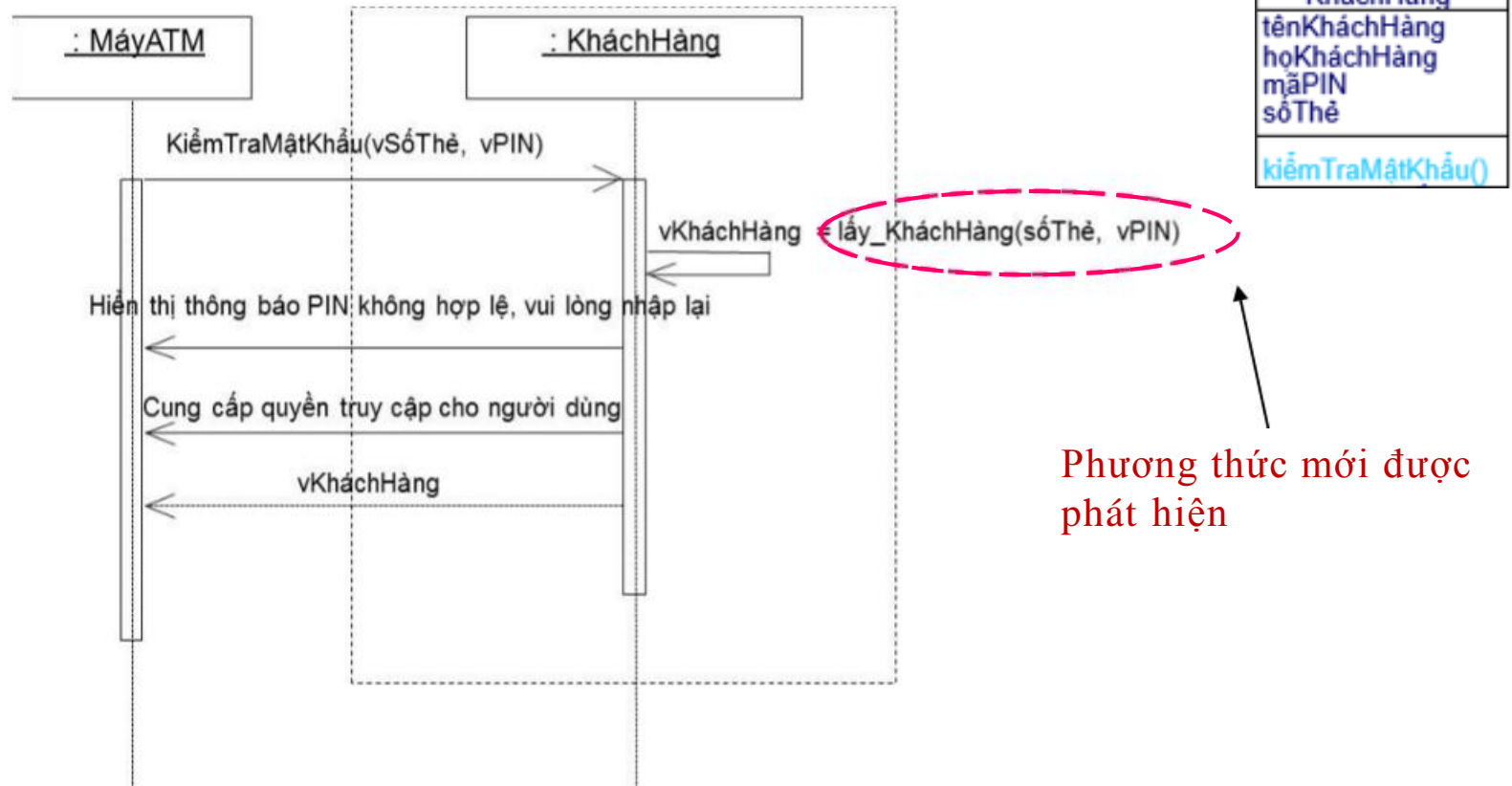
□ Ví dụ: Thiết kế phương thức hệ thống ATM

- `KháchHàng:: + kiểmTraMậtKhẩu (sốThẻ:String, vPIN:String) : vkháchHàng: KháchHàng`
- `TàiKhoản::+ rútTiền (sốTiền:float) :String`
- `TàiKhoản::+ gửiTiền (sốTiền:float) :String`
- `MáyATM::+ khởiĐộngMáy (sốTiềnKhởiTạo:float)`
- `MáyATM::+ đóngMáy ()`



Thiết kế lớp đối tượng: Tinh chế phương thức

- KháchHàng::+ kiểmTraMậtKhẩu (sốThẻ:String, vPIN:String) : vkháchHàng: KháchHàng



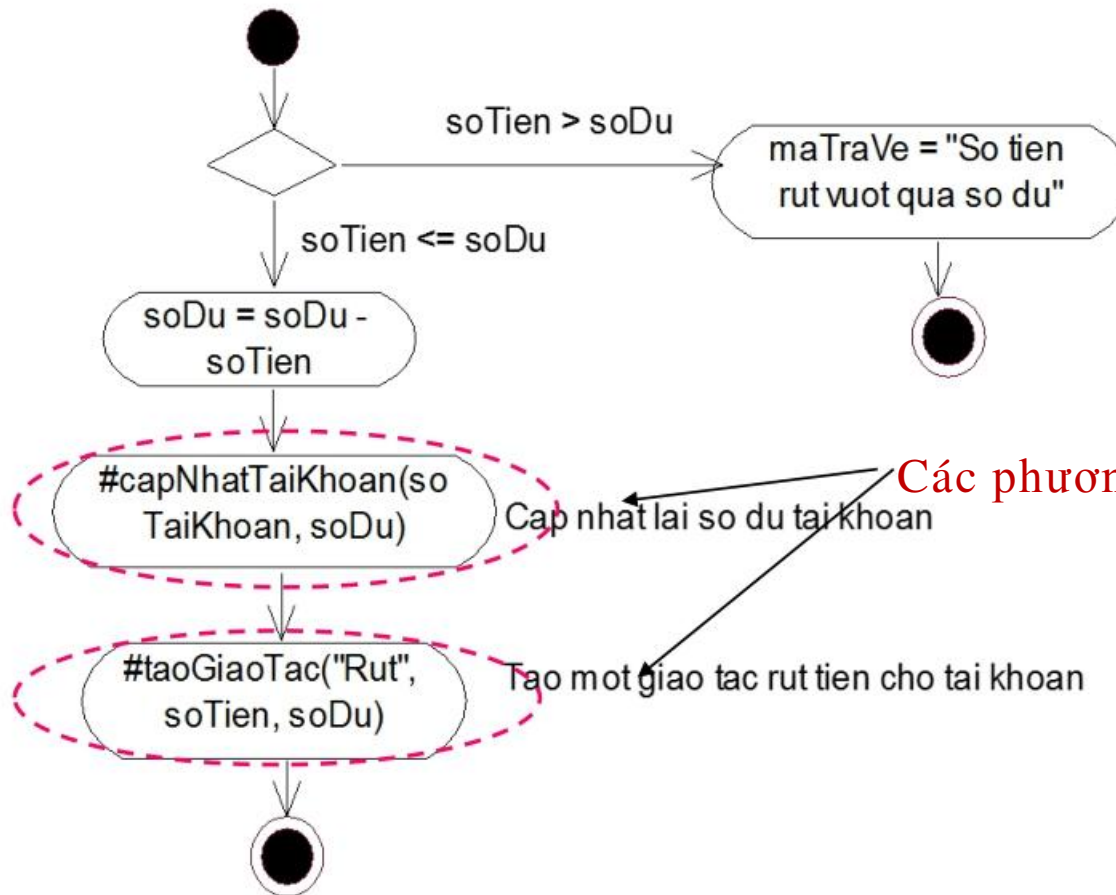
Thiết kế lớp đối tượng: Tinh chế phương thức

- ◉ `KháchHàng::+ kiểmTraMậtKhẩu (sốThẻ:String, vPIN:String) : vkháchHàng: KháchHàng`

Phương thức `kiểmTraMậtKhẩu()` trước hết sẽ thi hành tạo một đối tượng khách hàng và thực hiện lấy thông tin về khách hàng dựa trên dựa trên số thẻ và mã PIN. Để thực hiện điều này cần một phương thức là `lấy_KháchHàng()` và có phạm vi nội bộ dạng `protected (#)`. Phương thức này sẽ lấy tham số đầu vào là số thẻ và mã PIN, kết quả trả về là một đối tượng khách hàng tìm thấy hoặc là “null” nếu ngược lại. Nếu giá trị trả về là “null”, sẽ gửi một thông điệp tới hệ thống để thực hiện thông báo “PIN không hợp lệ, vui lòng nhập lại”, ngược lại, sẽ gán quyền truy cập cho người dùng.

Thiết kế lớp đối tượng: Tinh chế phương thức

⊙ TàiKhoản::+ rútTiền(*sốTiền: float*)



TàiKhoản
sốTàiKhoản
loạiTàiKhoản
sốDư
rútTiền() gửiTiền() xemTàiKhoản()

Các phương thức mới được phát hiện

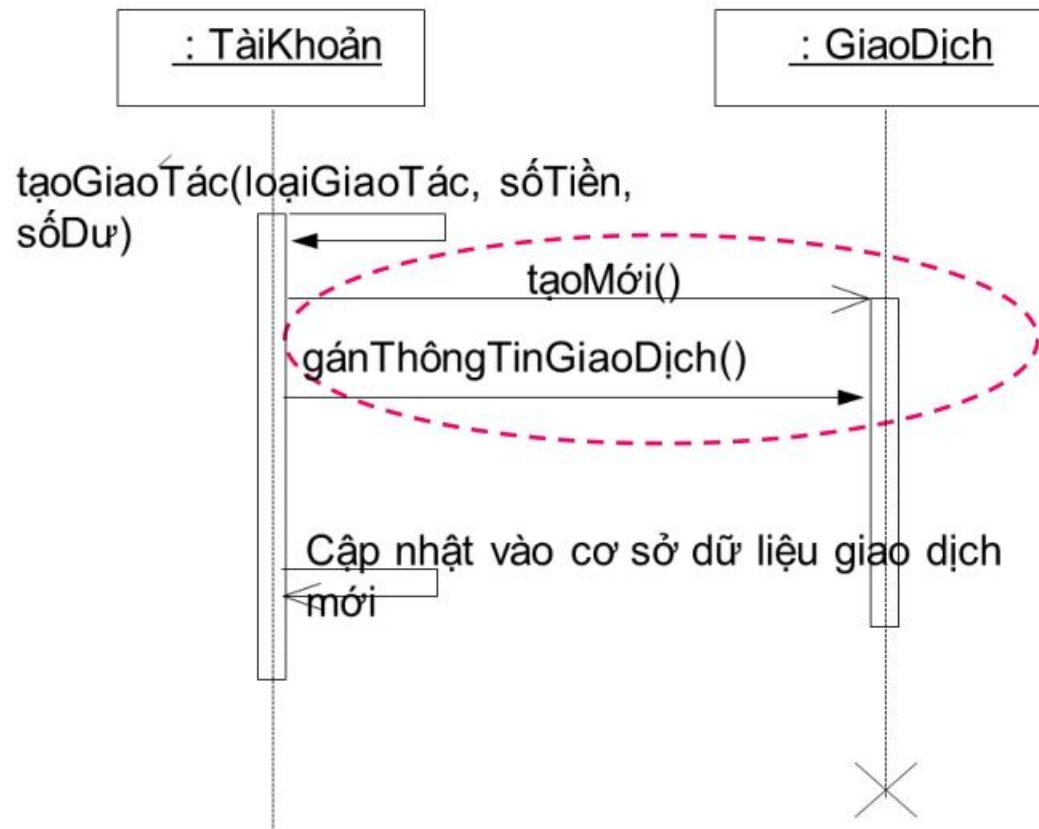
Thiết kế lớp đối tượng: Tinh chế phương thức

⦿ Tiếp tục tinh chế cho các phương thức mới phát hiện:

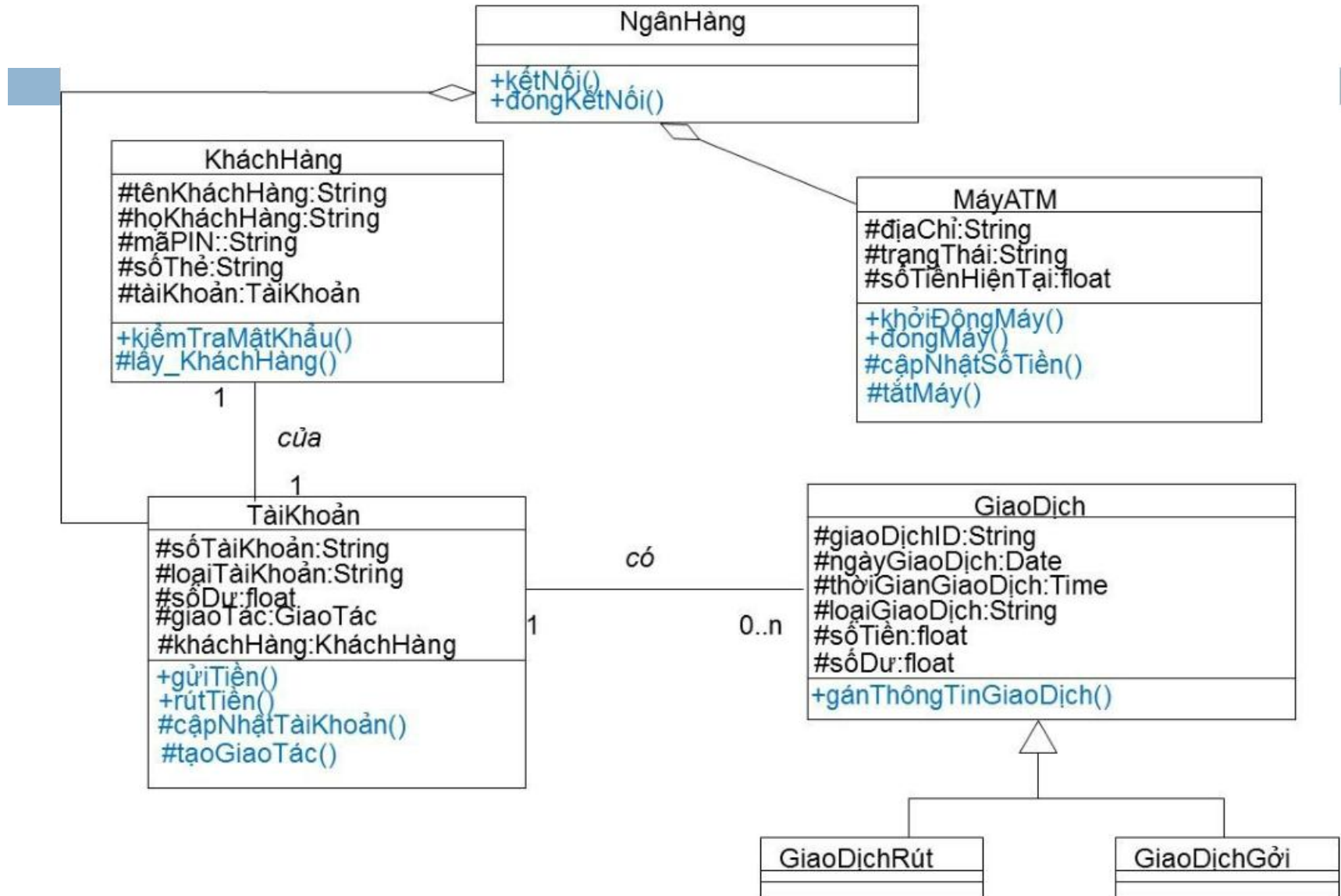
- KháchHàng:: *#lấy_TàiKhoản (sốThẻ, sốPIN)*
- TàiKhoản:: *#cậpNhậtTàiKhoản (sốTK, sốDư)*
- TàiKhoản:: *#tạoGiaoTác (loạiGT, sốTiền, sốDư)*

Thiết kế lớp đối tượng: Tinh chế phương thức

- `TàiKhoản::#tạoGiaoTắc(loạiGiaoTắc:String, sốTiền:float, sốDư:float)`



Thiết kế lớp đối tượng: Tinh chế phương thức



Nội dung

- Xác định các lớp đối tượng thông tin trong hệ thống phần mềm.
- Thiết kế các lớp đối tượng
- Chuyển đổi sơ đồ lớp sang mô hình dữ liệu quan hệ.
- Tối ưu lưu trữ đối tượng dựa trên Hệ QTCSDL quan hệ
- Ước lượng kích thước lưu trữ

Chuyển sơ đồ lớp sang mô hình dữ liệu quan hệ

- Chuyển đổi các lớp đối tượng sang các bảng dữ liệu
- Chuyển đổi các mối liên kết, bao gồm:
 - ⊙ Chuyển đổi liên kết kết hợp
 - ⊙ Chuyển đổi liên kết tổng quát hóa/kế thừa

Chuyển sơ đồ lớp sang mô hình dữ liệu quan hệ

□ Luật chuyển đổi

Luật 1: *Chuyển đổi tất cả các lớp đối tượng sang các bảng.* Nếu một lớp trong sơ đồ lớp có nhiều lớp con, chuyển đổi các lớp (bao gồm lớp cha, lớp con) sang các bảng trong lược đồ CSDL quan hệ (Luật 8).

Luật 2: *Chuyển đổi các thuộc tính đơn trị thành các thuộc tính của bảng dữ liệu.*

Luật 3: *Chuyển đổi các phương thức thành các thủ tục* (store procedure) hoặc các mô-đun chương trình.

Luật 4: *Chuyển đổi các mối kết hợp đơn trị (1-1)* bằng cách lấy thuộc tính khóa của một lớp này lưu trữ vào lớp còn lại, ví dụ như cách làm thêm khóa ngoại cho một bảng dữ liệu.

Chuyển sơ đồ lớp sang mô hình dữ liệu quan hệ

□ Luật chuyển đổi

Luật 5: *Chuyển đổi các thuộc tính đa trị* thành các bảng dữ liệu mới, và tạo ra mối quan hệ 1-n từ bảng gốc đến bảng mới.

Luật 6: *Chuyển đổi các mối kết hợp đa trị (n-n)* bằng cách tạo ra một bảng dữ liệu mới liên quan đến 2 bảng dữ liệu gốc. Lưu trữ các thuộc tính khóa chính của hai bảng dữ liệu gốc vào bảng dữ liệu như để làm khóa ngoại.

Luật 7: *Chuyển các mối kết hợp phối hợp (1-n)* bằng cách lấy thuộc tính khóa của lớp có giá trị đơn (1..1, 0..1) lưu vào lớp có giá trị nhiều (1..*, 0..*) thành thuộc tính như khóa ngoại.

Chuyển sơ đồ lớp sang mô hình dữ liệu quan hệ

□ Luật chuyển đổi

Luật 8: Sử dụng cho mỗi kết hợp tổng quát hóa/kế thừa:

Luật 8a: Sử dụng cả lớp cha (superclass) và lớp con (subclass).

Với trường hợp này thì phải đảm bảo khóa chính của lớp con giống như khóa chính của bảng cha, và xây dựng mối kết hợp 1-1 giữa lớp cha và lớp con.

Luật 8b: Sử dụng lớp con. Với trường hợp này chúng ta chuyển toàn bộ thuộc tính của lớp cha xuống lớp con, và thực hiện hủy lớp cha ra khỏi mô hình dữ liệu thiết kế.

Luật 8c: Sử dụng lớp cha. Với trường hợp này chúng ta chuyển toàn bộ thuộc tính của lớp con lên lớp cha, bổ sung thêm thuộc tính *Loại* để phân biệt giá trị của lớp con, sau đó thực hiện hủy lớp con ra khỏi mô hình thiết kế.

Chuyển đổi các lớp sang bảng dữ liệu

- ◉ Một lớp → một bảng (luật 1)
- ◉ Một thuộc tính đơn trị (bền vững) → một cột: chỉ có các thuộc tính có nhu cầu lưu trữ và được đòi hỏi bởi ứng dụng sẽ được chuyển thành cột của bảng (luật 2)
- ◉ Một đối tượng (thể hiện) → một dòng/bộ của bảng dữ liệu

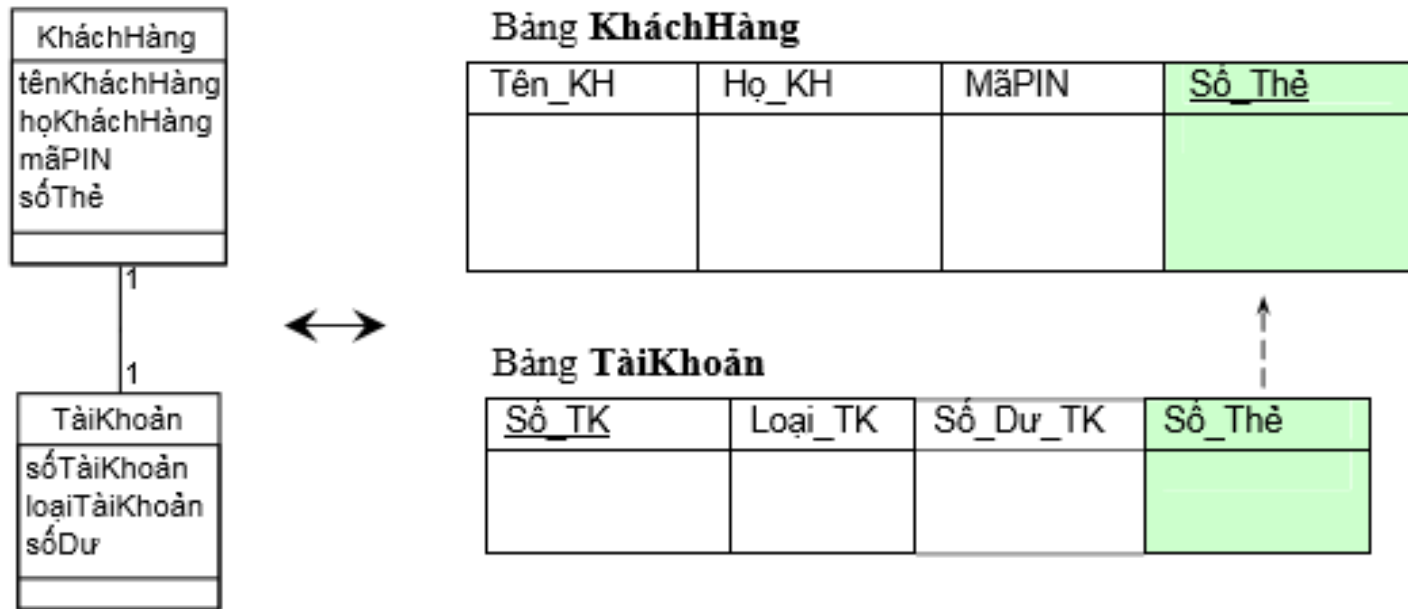
KháchHàng
tênKháchHàng
họKháchHàng
mãPIN
sốThẻ



Tên_KH	Họ_KH	MãPIN	<u>Số Thẻ</u>

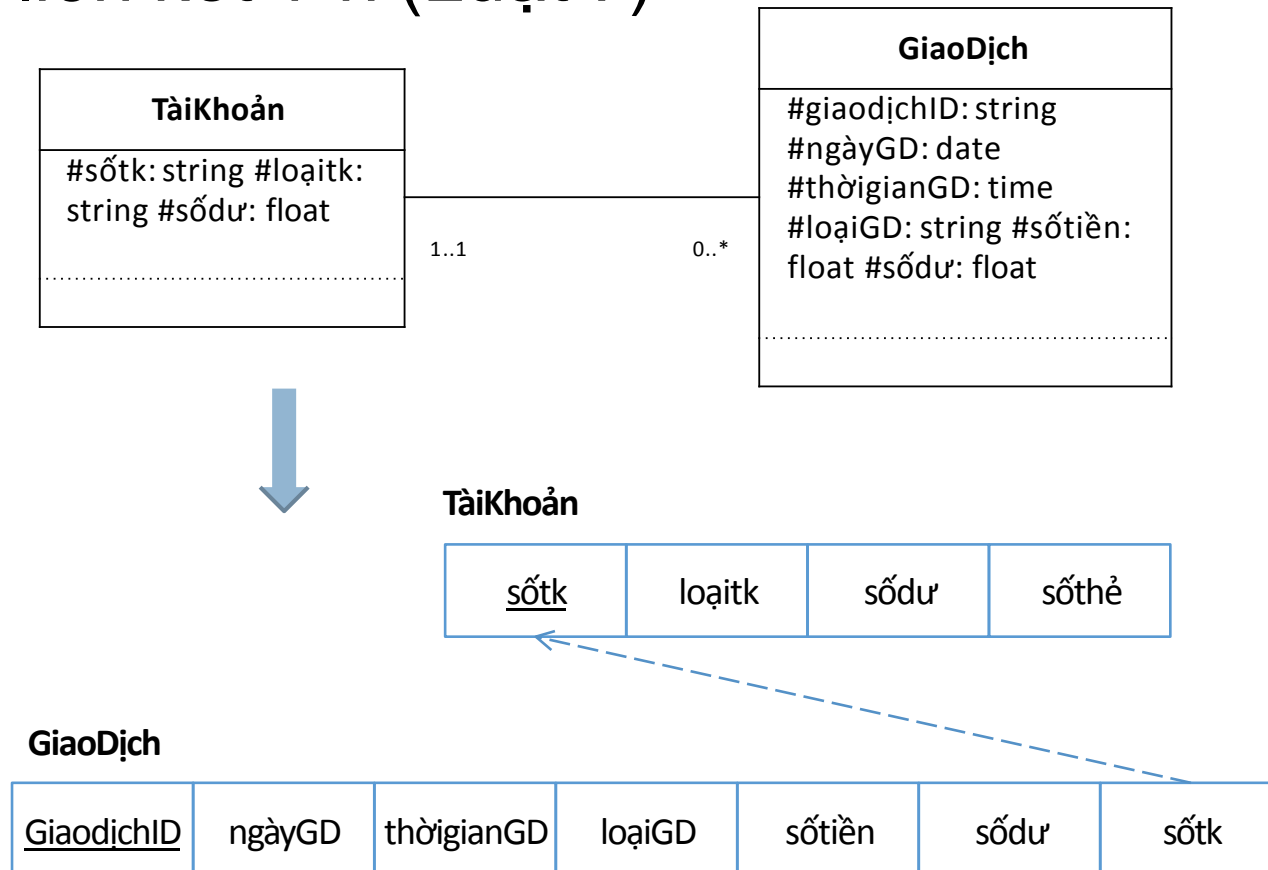
Chuyển đổi mối liên kết

□ Mối liên kết 1-1 (Luật 4)



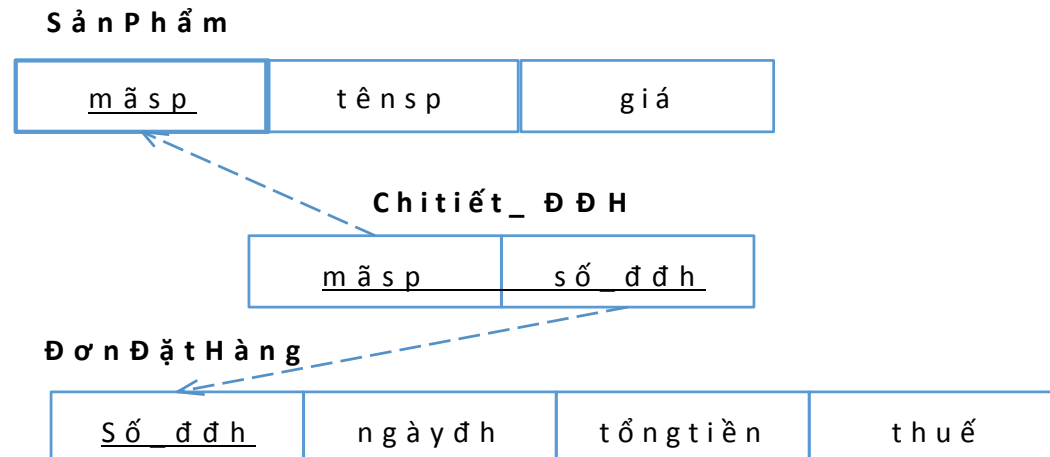
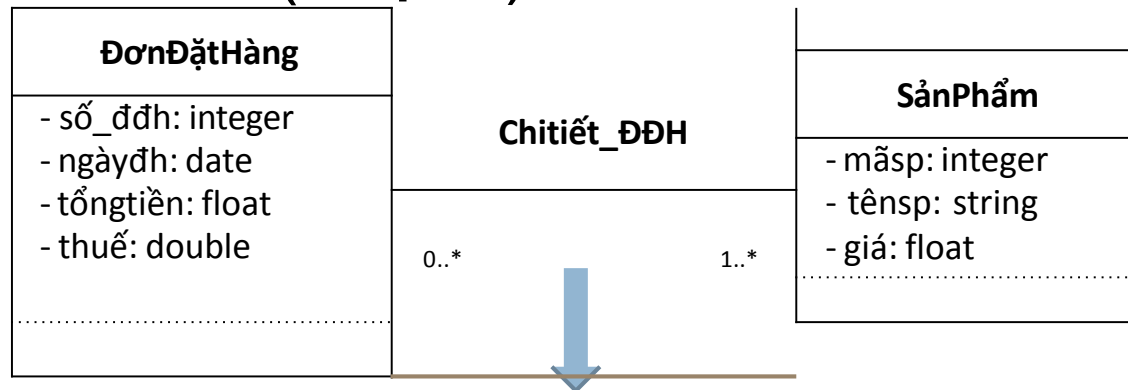
Chuyển đổi mối liên kết

□ Mối liên kết 1-n (Luật 7)



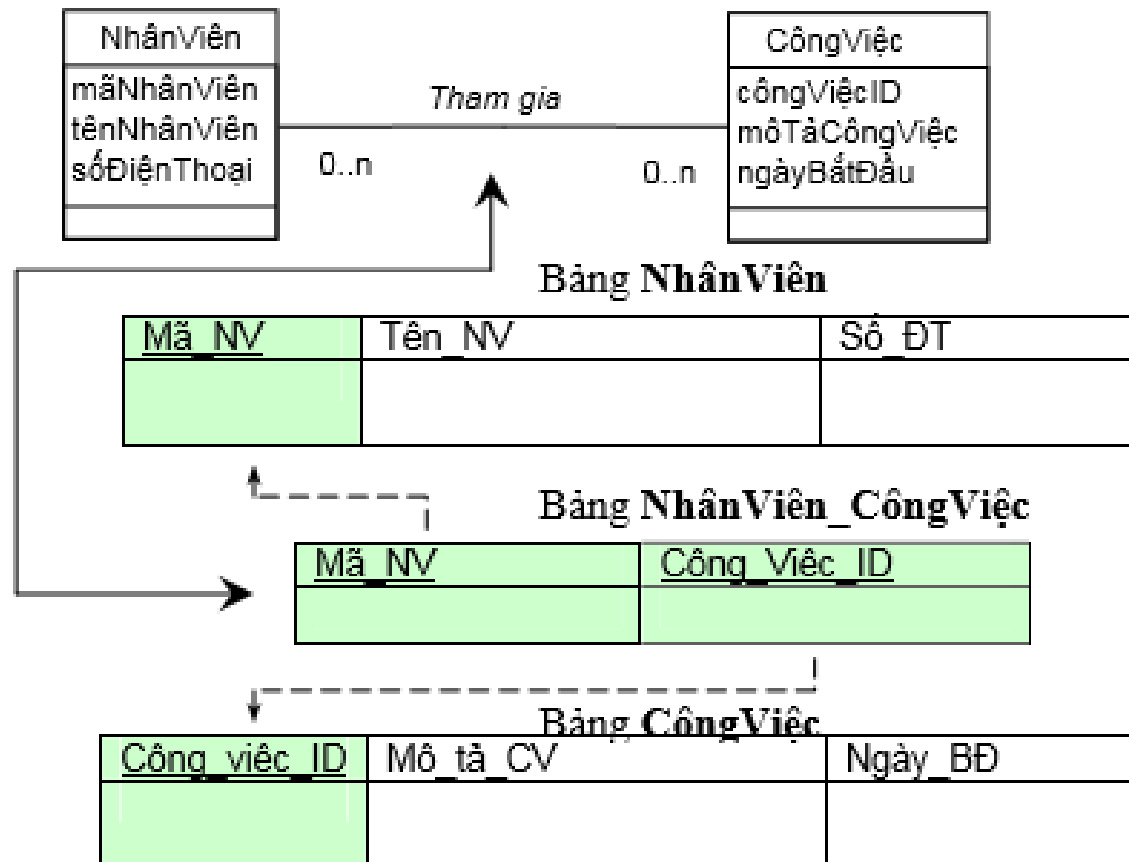
Chuyển đổi mối liên kết

□ Mối liên kết n-n (Luật 6)



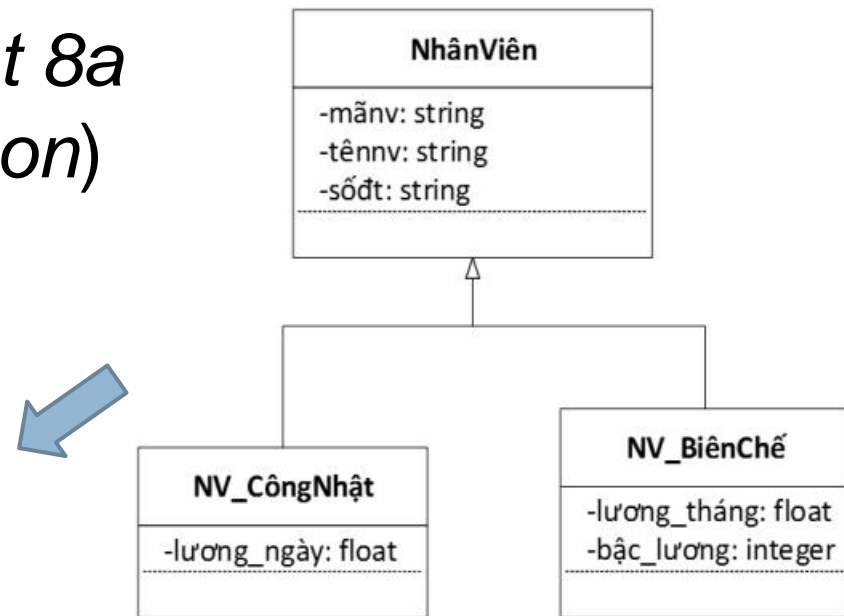
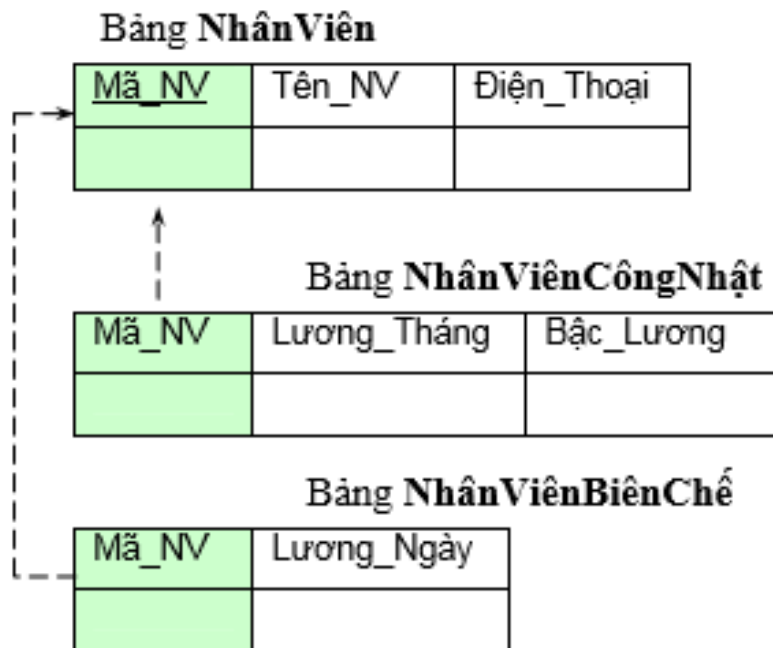
Chuyển đổi mối liên kết

□ Mối liên kết n-n (Luật 6)



Chuyển đổi mối liên kết

- Mối liên kết kế thừa (*Luật 8a sử dụng lớp cha và lớp con*)

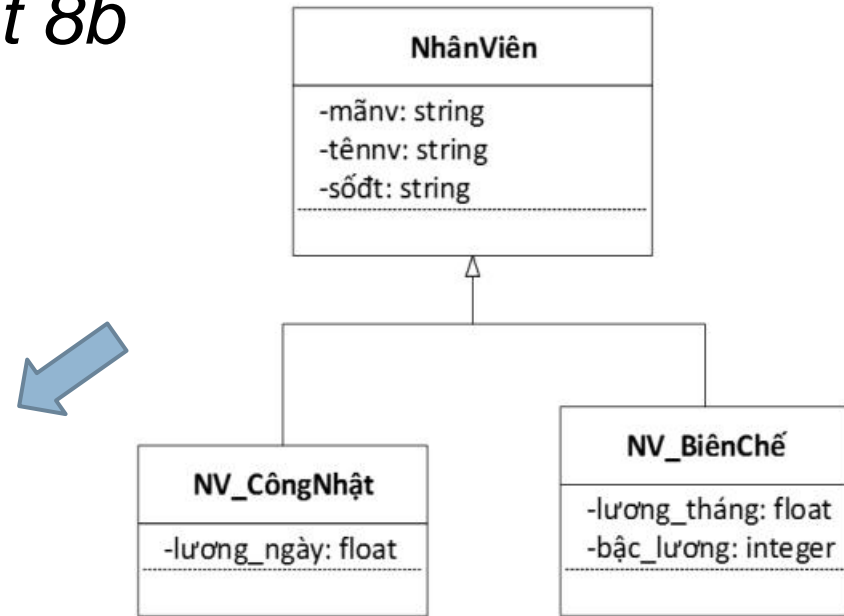


Hai bảng NV_CôngNhật và NV_BiênChế chỉ lưu khoá ngoại tham chiếu đến bảng nhân viên để tham khảo thông tin thừa kế.

Chuyển đổi mối liên kết

□ Mối liên kết kế thừa (*Luật 8b sử dụng lớp con*)

Chỉ dùng hai bảng NV_CôngNhật và NV_BiênChế. Tuy nhiên, tất cả các thuộc tính của lớp NhânViên sẽ được đưa vào hai bảng này. Nếu chúng ta muốn truy xuất thông tin về lớp NhânViên thì chúng ta có thể tạo một khung nhìn (view) hội của hai bảng này.



Bảng NhânViênCôngNhật

Mã_NV	Tên_NV	Điện_Thoại	Lương_Tháng	Bậc_Lương

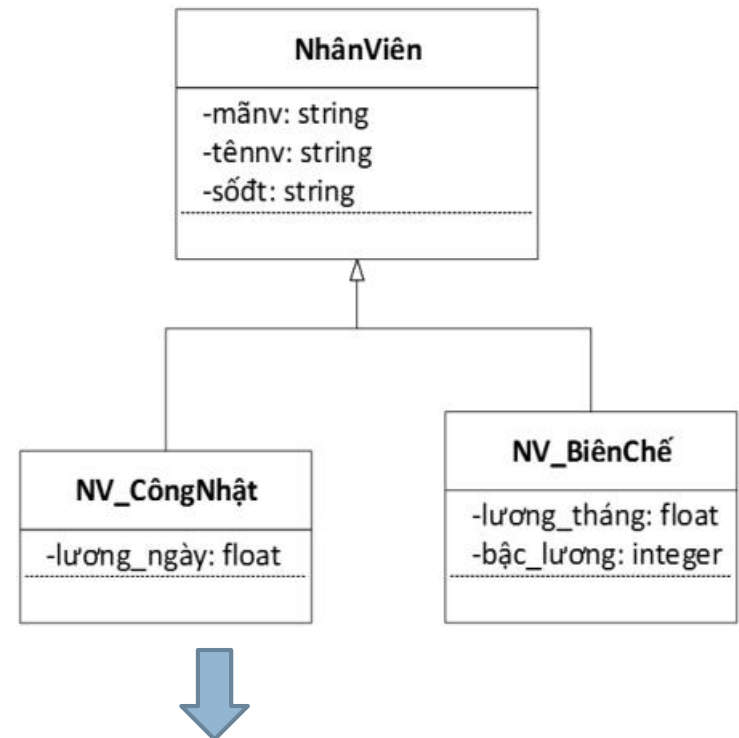
Bảng NhânViênBiênChế

Mã_NV	Tên_NV	Điện_Thoại	Lương_Ngày

Chuyển đổi mối liên kết

□ Mỗi liên kết kế thừa (Luật 8c sử dụng lớp cha)

Chỉ sử dụng một bảng lưu trữ tất cả các loại nhân viên. Do đó, các thuộc tính của bảng được hình thành từ các thuộc tính của lớp NhânViên, NV_CôngNhật và NV_BiênChế. Ngoài ra, đưa vào thêm một thuộc tính nhằm phân loại dòng này thuộc đối tượng của lớp nào.



Bảng NhânViên

Mã_NV	Tên_NV	Điện_Thoại	Lương_Ngày	Lương_Tháng	Bậc_Lương	Loại_NV

Nội dung

- Xác định các lớp đối tượng thông tin trong hệ thống phần mềm.
- Thiết kế các lớp đối tượng
- Chuyển đổi sơ đồ lớp sang mô hình dữ liệu quan hệ.
- Tối ưu lưu trữ đối tượng dựa trên HQT CSDL quan hệ
- Ước lượng kích thước lưu trữ

Tối ưu lưu trữ đối tượng dựa trên HQT CSDL quan hệ

- Tối ưu hóa về lưu trữ
 - ⊙ Sử dụng qui trình chuẩn hóa dữ liệu
- Tối ưu hóa về tốc độ truy cập
 - ⊙ Phi chuẩn hóa
 - ⊙ Dùng chỉ mục

Tối ưu hóa về lưu trữ: Quy trình chuẩn hóa dữ liệu

DẠNG CHUẨN 0

Có bảng dữ liệu nào có các thuộc tính bị lặp? Có bộ dữ liệu nào trong bảng có số thuộc tính khác với các bộ khác?

Có: Xóa các trường lặp. Thêm một bảng mới chứa các thuộc tính bị lặp.

Không: Lược đồ đạt DC1

DẠNG CHUẨN 1

Khóa chính của bảng dữ liệu bao gồm nhiều thuộc tính? Nếu có, có thuộc tính không khóa nào của bảng chỉ phụ thuộc vào một phần của khóa chính?

Có: Xóa phụ thuộc thành phần vào khóa. Thêm một bảng mới chứa các thuộc tính tạo ra phụ thuộc một phần vào khóa chính.

Không: Lược đồ đạt DC2

DẠNG CHUẨN 2

Có thuộc tính không khóa nào của bảng dữ liệu phụ thuộc lẫn nhau?

Có: Xóa các phụ thuộc giữa các thuộc tính không khóa. Thêm một bảng mới chứa các thuộc tính tạo ra phụ thuộc này.

Không: Lược đồ đạt DC3

DẠNG CHUẨN 3

Trùng lặp dữ liệu

Bảng VIII - 3. Minh họa lưu trữ trùng lặp thông tin

Bảng dữ liệu ĐƠN_HÀNG					
Mã_DH	Ngày_DH	Mã_KH	Tên_KH	Địa_chỉ	Điện_thoại
001	22/06/2015	1035	Nguyễn Thị Danh	135 Trần Hưng Đạo, Q5, Tp HCM	08.8354496
002	25/06/2015	1035	Nguyễn Thị Danh	135 Trần Hưng Đạo, Q5, Tp HCM	08.8354496
003	27/06/2015	1035	Nguyễn Thị Danh	135 Trần Hưng Đạo, Q5, Tp HCM	08.8354496
004	20/06/2015	1036	Phan Đình Hùng	246 CMT8, Q.TB, Tp HCM	08.8375963
005	15/05/2015	1036	Phan Đình Hùng	246 CMT8, Q.TB, Tp HCM	08.8375963
006	30/05/2015	1036	Phan Đình Hùng	246 CMT8, Q.TB, Tp HCM	08.8375963
007	14/04/2015	1037	Mai Văn Hương	257 Võ Thị Sáu, Q1, Tp HCM	NULL
008	19/04/2015	1037	Mai Văn Hương	257 Võ Thị Sáu, Q1, Tp HCM	NULL
009	25/04/2015	1037	Mai Văn Hương	257 Võ Thị Sáu, Q1, Tp HCM	NULL

Lưu trữ giá trị NULL

Bảng VIII - 4. Minh họa dạng chuẩn 0

Bảng dữ liệu THUÊ_NHÀ (MÃ_NT, Tên_NT, (Mã_Nhà, ĐC_Nhà, BD_Thuê, KT_Thuê, Giá_thuê, Mã_CN, Tên_CN)*n)								
Mã_NT	Tên_NT	Mã_Nhà	ĐC_Nhà	BD_Thuê	KT_Thuê	Giá_thuê	Mã_CN	Tên_CN
CR76	N.T.A	PG4	123 CMT8, Q3, TpHCM	1/1/2013	1/1/2014	350	CO40	T.V.H
		PG16	23 Trương Định, Q1, Tp HCM	1/1/2014	1/1/2015	450	CO93	H.V.B
CR56	B.V.H	PG4	123 CMT8, Q3, TpHCM	1/1/2012	1/1/2013	350	CO40	T.V.H
		PG16	23 Trương Định, Q1, Tp HCM	1/1/2013	1/1/2014	450	CO93	H.V.B
		PG36	85 Trần Phú, Q5, Tp HCM	1/1/2014	1/1/2015	500	CO93	H.V.B

Mỗi người thuê thuê n căn nhà, như vậy thông tin căn nhà thuê (Mã_Nhà, ĐC_Nhà, BD_Thuê, KT_Thuê, Giá_thuê, Mã_CN, Tên_CN) lặp lại trên mỗi bộ tương ứng với số nhà mà mỗi người thuê thuê, ví dụ CR76 lặp 2 lần, CR56 lặp 3 lần

Bảng VIII - 5. Minh họa dạng chuẩn 1

Bảng dữ liệu THUÊ_NHÀ (Mã_NT, Tên_NT, Mã_Nhà, ĐC_Nhà, BD_Thuê, KT_Thuê, Giá_thuê, Mã_CN, Tên_CN)								
Mã_NT	Tên_NT	Mã_Nhà	ĐC_Nhà	BD_Thuê	KT_Thuê	Giá_thuê	Mã_CN	Tên_CN
CR76	N.T.A	PG4	123 CMT8, Q3, TpHCM	1/1/2013	1/1/2014	350	CO40	T.V.H
CR76	N.T.A	PG16	23 Trương Định, Q1, Tp HCM	1/1/2014	1/1/2015	450	CO93	H.V.B
CR56	B.V.H	PG4	123 CMT8, Q3, TpHCM	1/1/2012	1/1/2013	350	CO40	T.V.H
CR56	B.V.H	PG16	23 Trương Định, Q1, Tp HCM	1/1/2013	1/1/2014	450	CO93	H.V.B
CR56	B.V.H	PG36	85 Trần Phú, Q5, Tp HCM	1/1/2014	1/1/2015	500	CO93	H.V.B

Bảng VIII - 6. Minh họa mô hình dữ liệu ở dạng chuẩn 2



Bảng dữ liệu NGƯỜI THUÊ NGƯỜI_THUÊ (Mã_NT, Tên_NT)	
Mã_NT	Tên_NT
CR76	N.T.A
CR56	B.V.H



Bảng dữ liệu THUÊ_NHÀ THUÊ_NHÀ(Mã_NT, Mã_Nhà, BD_Thuê, KT_Thuê)			
Mã_NT	Mã_Nhà	BD_Thuê	KT_Thuê
CR76	PG4	1/1/2013	1/1/2014
CR76	PG16	1/1/2014	1/1/2015
CR56	PG4	1/1/2012	1/1/2013
CR56	PG16	1/1/2013	1/1/2014
CR56	PG36	1/1/2014	1/1/2015

Bảng dữ liệu NHÀ_THUÊ NHÀ_THUÊ (Mã_Nhà, ĐC_Nhà, Giá_thuê, Mã_CN, Tên_CN)				
Mã_Nhà	ĐC_Nhà	Giá_thuê	Mã_CN	Tên_CN
PG4	123 CMT8, Q3, TpHCM	350	CO40	T.V.H
PG16	23 Trương Định, Q1, Tp HCM	450	CO93	H.V.B
PG36	85 Trần Phú, Q5, Tp HCM	500	CO93	H.V.B

Bảng VIII - 7. Minh họa mô hình dữ liệu đạt đạt dạng chuẩn 3

Bảng dữ liệu CHỦ_NHÀ NHÀ_THUÊ (Mã_CN, Tên_CN)	
Mã_CN	Tên_CN
CO40	T.V.H
CO93	H.V.B
CO93	H.V.B

Bảng dữ liệu THUÊ_NHÀ THUÊ_NHÀ(Mã_NT, Mã_Nhà, BD_Thuê, KT_Thuê)			
Mã_NT	Mã_Nhà	BD_Thuê	KT_Thuê
CR76	PG4	1/1/2013	1/1/2014
CR76	PG16	1/1/2014	1/1/2015
CR56	PG4	1/1/2012	1/1/2013
CR56	PG16	1/1/2013	1/1/2014
CR56	PG36	1/1/2014	1/1/2015

Bảng dữ liệu NHÀ_THUÊ NHÀ_THUÊ (Mã_Nhà, ĐC_Nhà, Giá_thuê, Mã_CN)			
Mã_Nhà	ĐC_Nhà	Giá_thuê	Mã_CN
PG4	123 CMT8, Q3, TpHCM	350	CO40
PG16	23 Trương Định, Q1, Tp HCM	450	CO93
PG36	85 Trần Phú, Q5, Tp HCM	500	CO93

Bảng dữ liệu NGƯỜI THUÊ NGƯỜI_THUÊ (Mã_NT, Tên_NT)	
Mã_NT	Tên_NT
CR76	N.T.A
CR56	B.V.H

Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

□ Lược đồ CSDL cho thuê nhà

- 1) Client (clientNo, fname, lname, telNo, maxRent)
- 2) Interview (clientNo, staffNo, dateInterview, comment)
- 3) Viewing (clientNo, propertyNo, viewDate, comment)
- 4) PropertyForRent (propertyNo, street, city, postcode, rooms, rent, ownerNo, staffNo, branchNo)
- 5) Owner (ownerNo, fname, lname, address, telNo)
- 6) Branch (branchNo, street, city, postcode)
- 7) TelephoneBranch (telNo, branchNo)
- 8) PropertyType (type, description)

Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

- ❑ Lưu trữ trùng lặp thuộc tính trên mỗi kết hợp **1-1**
- ❑ Ví dụ: gộp 2 quan hệ Client và Interview thành một quan hệ ClientInterview
 - 1) Client(clientNo, fname, lname, telNo, maxRent)
 - 2) Interview(clientNo, staffNo, dateInterview, comment)

ClientInterview (clientNo, fname, lname, telNo, maxRent, staffNo, dateInterview, comment) . . .

Lưu ý giá
trị NULL

Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

- ❑ Lưu trữ trùng lặp thuộc tính không khóa trên mỗi kết hợp **1-n**
- ❑ Ví dụ: Cho biết thông tin nhà cho thuê và tên ông chủ nhà
 - 1) PropertyForRent(propertyNo, street, city, postcode, rooms, rent, ownerNo, staffNo, branchNo)
 - 2) Owner(ownerNo, fname, lname, address, telNo)

PropertyForRent

<u>propertyNo</u>	OwnerNo	lName	staffNo	branchNo
-------------------	-----	-----	-----	-----	---------	-------	---------	----------

Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

- ❑ Lưu trữ trùng lặp thuộc tính khóa trên mỗi kết hợp 1-n
- ❑ Ví dụ: Cho biết thông tin tất cả các chủ nhà liên quan đến chi nhánh B003
 - 1) PropertyForRent(propertyNo, street, city, postcode, rooms, rent, ownerNo, staffNo, branchNo)
 - 2) Owner(ownerNo, fname, lname, address, telNo)

Owner

<u>OwnerNo</u>	branchNo
----------------	-----	-----	-----	-----	----------

Chủ nhà
thuộc về
nhiều chi
nhánh?

Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

- ❑ Lưu trữ trùng lặp thuộc tính trên mỗi kết hợp **n-n**
- ❑ Ví dụ: Cho biết thông tin khách xem nhà (mã người thuê, mã căn nhà, địa chỉ căn nhà)
 - 1) Client(clientNo, fname, lname, telNo, maxRent)
 - 2) Viewing(clientNo, propertyNo, viewDate, comment)
 - 3) PropertyForRent(propertyNo, street, city, postcode, rooms, rent, ownerNo, staffNo, branchNo)

Viewing

<u>clientNo</u>	<u>propertyNo</u>	street	viewDate	comment
-----------------	-------------------	--------	----------	---------

Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

- Lưu trữ theo nhóm lặp
- Ví dụ: Giả sử mỗi chi nhánh có tối đa 3 điện thoại \Rightarrow gộp bảng điện thoại vào chi nhánh
 - 1) Branch(branchNo, street, city, postcode)
 - 2) TelephoneBranch(telNo, branchNo)

Branch

branchNo	street	city	postcode	telNo1	telNo2	telNo1
----------	--------	------	----------	--------	--------	--------

Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

- Gộp quan hệ
- Ví dụ: Giả sử có truy xuất dữ liệu thường xuyên truy xuất trên thông tin nhà và mô tả loại nhà với tần xuất rất lớn \Rightarrow gộp bảng nhà cho thuê và loại nhà
 - 1) PropertyForRent (propertyNo, street, city, postcode, rooms, rent, ownerNo, staffNo, branchNo, type)
 - 2) PropertyType (type, description)

PropertyForRent

propertyNo	type	discription	OwnerNo	lName	staffNo	branchNo
------------	-----	-----	------	-------------	---------	-------	---------	----------

Tối ưu hóa về tốc độ truy cập: dùng chỉ mục

□ Các lưu ý cho kỹ thuật chỉ mục:

Hạn chế sử dụng chỉ mục cho hệ thống giao tác (transaction systems).

Sử dụng nhiều chỉ mục để giảm thời gian hồi đáp dữ liệu trong các hệ thống hỗ trợ ra quyết định.

Trong mỗi bảng dữ liệu, nên tạo chỉ mục duy nhất (unique index) cho tập thuộc tính khóa chính.

Trong mỗi bảng dữ liệu, nên tạo chỉ mục cho thuộc tính khóa ngoại để tăng hiệu quả phép kết (join).

Tạo chỉ mục cho các thuộc tính mà được sử dụng thường xuyên trong gom nhóm (grouping), sắp xếp (sorting) và điều kiện tìm kiếm (criteria).

Tối ưu hóa về tốc độ truy cập: dung chỉ mục

□ Các lưu ý cho kỹ thuật dùng chỉ mục:

Không chỉ mục trên các quan hệ nhỏ

Nên thêm chỉ mục cho bất kỳ thuộc tính nào thường xuyên truy cập

Tránh chỉ mục trên các quan hệ/thuộc tính thường xuyên cập nhật.

Tránh chỉ mục trên thuộc tính nếu truy vấn trả về nhiều hơn 25% số bộ của quan hệ.

Tránh chỉ mục trên thuộc tính có kiểu chuỗi dài