

Chương IV: Phân tích sơ đồ lớp

4.1. Biểu đồ lớp

a. Ý nghĩa

Trong phương pháp hướng đối tượng, *một nhóm đối tượng có chung một số thuộc tính và phương thức tạo thành một lớp*. Mỗi tương tác giữa các đối tượng trong hệ thống sẽ được biểu diễn thông qua mối quan hệ giữa các lớp.

Các lớp (bao gồm cả các thuộc tính và phương thức) cùng với các mối quan hệ sẽ tạo thành biểu đồ lớp. Biểu đồ lớp là một biểu đồ dạng mô hình tĩnh nhằm mô tả hướng nhìn tĩnh về một hệ thống bằng các khái niệm lớp, các thuộc tính, phương thức của lớp và mối quan hệ giữa chúng với nhau.

b. Tập ký hiệu UML cho biểu đồ lớp

- i. **Kí hiệu lớp:** trong UML, mỗi lớp được biểu diễn bởi hình chữ nhật gồm 3 phần: tên lớp, các thuộc tính và các phương thức.
- ii. **Thuộc tính:** các thuộc tính trong biểu đồ lớp được biểu diễn theo cấu trúc chung như sau:

phạm_vi tên : kiểu = mặc_định

Trong đó:

- *phạm_vi*: cho biết phạm vi truy nhập của thuộc tính. Có ba kiểu xác định thuộc tính phổ biến là:

+: thuộc tính kiểu public

#: thuộc tính kiểu protected

-: thuộc tính kiểu private.

Các phạm vi của thuộc tính có thể được biểu diễn dưới dạng ký hiệu (+, #, -) hoặc biểu diễn dưới dạng các từ khoá (public, protected, private).

- *Tên*: là xâu ký tự biểu diễn tên thuộc tính.

- *kiểu*: là kiểu dữ liệu của thuộc tính.

- *mặc_định*: là giá trị khởi đầu mặc định (nếu có) của thuộc tính.

- iii. **Phương thức (method):** các phương thức trong UML được biểu diễn theo cấu trúc chung như sau:

phạm_vi_tên(danh_sách_tham_số): kiểu_trả_lại { ki_êu_ph ương_thức }

Trong đó:

- *phạm_vi* biểu diễn phạm vi cho phương thức. Giống như đối với thuộc tính, có ba dạng kiểu xác định cơ bản cho phương thức là:
 - +: phương thức kiểu public
 - #: phương thức kiểu protected
 - : phương thức kiểu private
 - *tên* là xâu ký tự xác định tên của phương thức.
 - *kiểu_trả_về*: chỉ ra kiểu giá trị trả về của phương thức.
 - *danh_sách_tham_số*: biểu diễn danh sách các tham số trong khai báo của phương thức. Mỗi tham số được biểu diễn dưới dạng chung:

tên tham số: *kiểu giá trị = giá trị mặc định*.
 - *ki_êu_ph ương_thức*: không bắt buộc, cho biết kiểu phương thức.
- Phương thức có thể nhận một trong các kiểu đặc biệt sau:

abstract: phương thức kiểu trừu tượng

Ví dụ một khai báo phương thức cho một lớp:

generatePurchaseList(prodID:int): String

4.2. Các mối quan hệ trong biểu đồ lớp

Giữa các lớp có các dạng quan hệ cơ bản như sau:

a. Quan hệ kết hợp (Association):

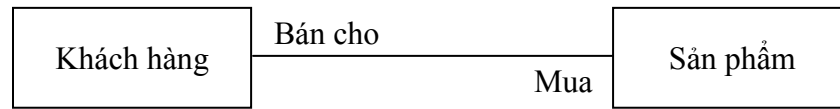
Một kết hợp (association) là một sự nối kết giữa các lớp, cũng có nghĩa là sự nối kết giữa các đối tượng của các lớp này.

Trong UML, một quan hệ được xác định nhằm mô tả một tập hợp các liên kết (links), tức là một sự liên quan về ngữ nghĩa (semantic connection) giữa một nhóm các đối tượng được biểu diễn bởi các lớp tương ứng.

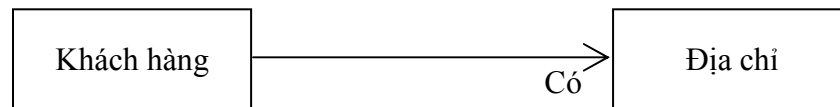
Mặc định, quan hệ kết hợp được biểu diễn bởi đoạn thẳng 2 chiều nối 2 đối tượng và có thể kèm theo ngữ nghĩa của quan hệ tại hai đầu của đoạn thẳng.

Ví dụ: Lớp khách hàng có quan hệ kết hợp với lớp sản phẩm.

Ngữ nghĩa của quan hệ này thể hiện ở chỗ: khách hàng *mua* sản phẩm, còn sản phẩm *được bán cho* khách hàng.

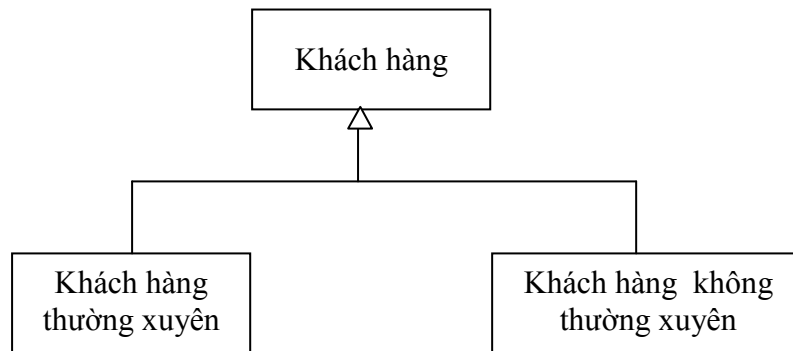


Quan hệ kết hợp còn có dạng một chiều:



b. **Khái quát hóa (Generalization):**

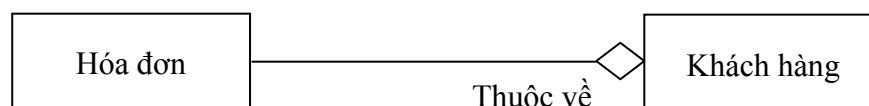
Khái quát hóa là mối quan hệ giữa một lớp có các đặc trưng mang tính khái quát cao hơn và một lớp có tính chất đặc biệt hơn. Trong sơ đồ lớp, mối quan hệ khái quát hóa chính là sự kế thừa của một lớp từ lớp khác. Quan hệ khái quát hoá được biểu diễn bằng một mũi tên có tam giác rỗng gắn ở đầu



c. **Quan hệ cộng hợp (Aggregation):**

là dạng quan hệ mô tả một lớp A là một phần của lớp B và lớp A có thể tồn tại độc lập. Quan hệ cộng hợp được biểu diễn bằng một mũi tên gắn hình thoi rỗng ở đầu hướng về lớp bao hàm.

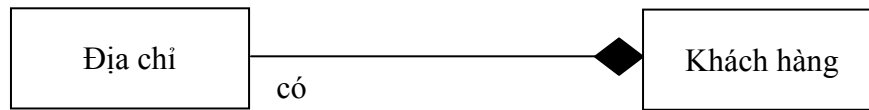
Xem ví dụ. Lớp Hoá đơn là một phần của lớp Khách hàng nhưng đối tượng Hoá đơn vẫn có thể tồn tại độc lập với đối tượng khách hàng.



d. **Quan hệ gộp (Composition):**

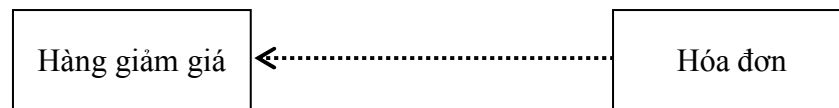
Một quan hệ gộp biểu diễn một quan hệ kiểu tổng thể-bộ phận. Lớp A có quan hệ gộp với lớp B nếu lớp A là một phần của lớp B và sự tồn tại của đối

tượng lớp B điều khiển sự tồn tại của đối tượng lớp A. Quan hệ này được biểu diễn bởi một mũi tên gắn hình thoi đặc ở đầu.



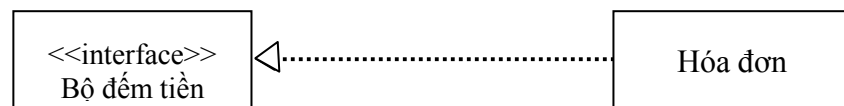
e. **Quan hệ phụ thuộc (Dependency):**

Phụ thuộc là mối quan hệ giữa hai lớp đối tượng: một lớp đối tượng A có tính độc lập và một lớp đối tượng B phụ thuộc vào A; một sự thay đổi của A sẽ ảnh hưởng đến lớp phụ thuộc B.



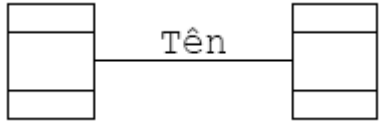
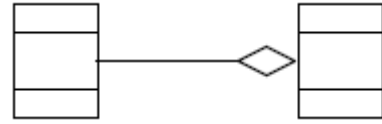


f. **Quan hệ thực thi (Realization):**

Biểu diễn mối quan hệ ngữ nghĩa giữa các thành phần của biểu đồ lớp, trong đó một thành phần mô tả một công việc dạng hợp đồng và thành phần còn lại thực hiện hợp đồng đó. Thông thường lớp thực hiện hợp đồng có thể là các giao diện

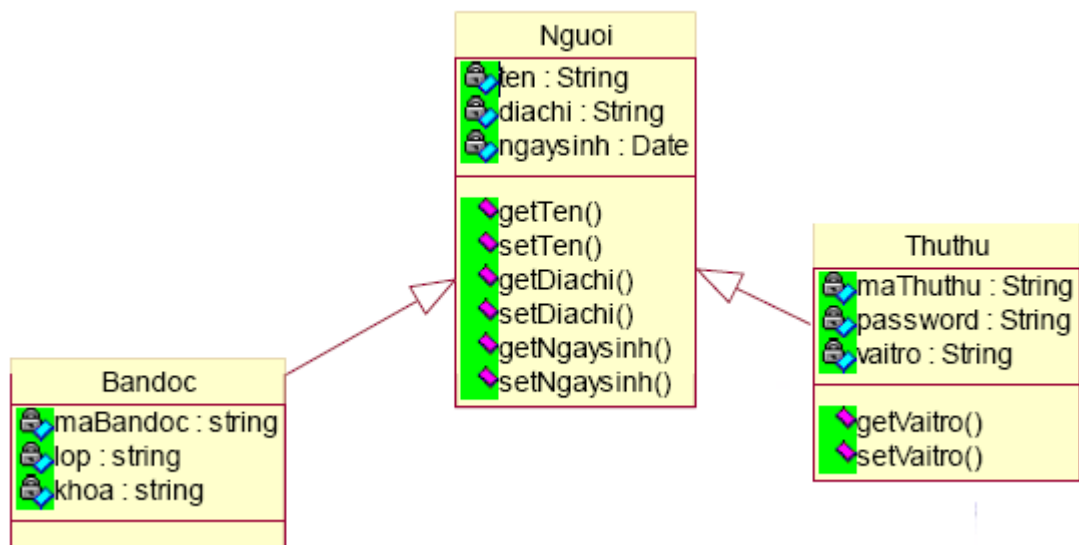


Tổng kết các phần tử mô hình UML được sử dụng trong mô hình lớp, ý nghĩa và ký hiệu tương ứng trong các biểu đồ.

Phần tử mô hình	Ý nghĩa	Cách biểu diễn	Ký hiệu trong biểu đồ
Lớp (class)	Biểu diễn tên lớp, các thuộc tính và phương thức của lớp đó.	Một hình chữ nhật gồm 3 phần tách biệt.	<div style="border: 1px solid black; padding: 5px; margin: 5px;"> <div style="border-bottom: 1px solid black; padding: 2px 5px;">Tên lớp</div> <div style="border-bottom: 1px solid black; padding: 2px 5px;">Các thuộc tính</div> <div style="padding: 2px 5px;">Các phương thức</div> </div>

Quan hệ kiểu kết hợp	Biểu diễn quan hệ giữa hai lớp độc lập, có liên quan đến nhau.	Một đường kẻ liền nét (có tên xác định) nối giữa hai lớp.	
Quan hệ gộp	Biểu diễn quan hệ kiểu bộ phận – tổng thể.	Đường kẻ liền nét có hình thoi ở đầu.	
Quan hệ khái quát hoá (kế thừa)	Lớp này thừa hưởng các thuộc tính - phương thức của lớp kia	Mũi tên tam giác.	
Quan hệ phụ thuộc.	Các lớp phụ thuộc lẫn nhau trong hoạt động của hệ thống.	Mũi tên đứt nét.	

Dưới đây là ví dụ một phần của biểu đồ lớp trong hệ thống quản lý thư viện trong đó các lớp Thủ thư (người quản lý thư viện) và Bạn đọc kế thừa từ lớp Person.



Chương V: Thư viện chuẩn STL

(Standard Template Library)

5.1. **Giới thiệu:**

Trong C++, **Thư viện chuẩn** hay **Thư viện tiêu chuẩn** là một tập hợp của các lớp và các hàm. Thư viện chuẩn này cung cấp nhiều thùng chứa (container) tổng quát, các hàm để làm tiện ích và điều chỉnh cho các thùng chứa, các đối tượng hàm, các dãy kí tự tổng quát và các dòng dữ liệu (bao gồm I/O tương tác và tập tin), hỗ trợ một số tính năng ngôn ngữ, và nhiều hàm thông dụng. Các thư viện chuẩn C++ khai báo trong không gian tên *std*.

Các tập tin sau đây chứa các khai báo của Thư viện chuẩn và miêu tả chung của chúng:

a. **Các thùng chứa**

<bitset> -- Định nghĩa khuôn mẫu lớp bitset và các phương thức hỗ trợ cho việc hiển thị và điều chỉnh các dãy bit có độ dài cố định.

<deque> -- Định nghĩa khuôn mẫu lớp deque (hàng).

<list> -- Định nghĩa khuôn mẫu lớp list (danh sách)

<map> -- Định nghĩa khuôn mẫu lớp map và <multimap>

<queue> -- Định nghĩa khuôn mẫu lớp priority_queue và queue

<set> -- Định nghĩa khuôn mẫu lớp set

<stack> -- Định nghĩa khuôn mẫu lớp stack (chồng)

<vector> -- Định nghĩa khuôn mẫu lớp vector

b. **Tổng quát:**

<algorithm> -- Định nghĩa nhiều hàm thực thi các thuật toán về thùng chứa trong Thư viện template chuẩn STL.

<functional> -- Định nghĩa các hàm nhằm giúp cho việc xây dựng các đối tượng hàm trong Thư viện khuôn mẫu chuẩn (còn có tên là *hàm tử*).

<iterator> -- Định nghĩa các biến lặp cơ bản, định nghĩa sẵn các biến lặp, biến lặp dòng.

<locale> -- Định nghĩa các lớp khuôn mẫu và các hàm mà chương trình C++ có thể dùng để tạo đặc tính đóng và điều chỉnh các yếu tố định dạng và biểu thị của các số, tiền tệ, dữ liệu về ngày tháng bao gồm hỗ trợ ngôn ngữ quốc tế cho sự phân lớp kí tự và sự sắp xếp dãy kí tự.

<memory> -- Định nghĩa một lớp, một toán tử và nhiều khuôn mẫu nhằm giúp cấp phát và trả về (vùng nhớ) của các đối tượng.

<stdexcept> -- Định nghĩa nhiều lớp chuẩn dùng cho việc báo cáo các ngoại lệ. Các lớp tạo nên một hệ phân nhánh mà tất cả đều dẫn xuất từ lớp exception và bao gồm hai kiểu ngoại lệ thông dụng: lỗi lô-gíc và lỗi trong thời gian thi hành.

<utility> -- Định nghĩa các kiểu, hàm, và toán tử trong Thư viện khuôn mẫu chuẩn STL nhằm giúp cho việc xây dựng và quản lý các cặp đối tượng. Nó hữu dụng, khi hai đối tượng cần được hành xử như chúng là một.

c. **Các dãy kí tự:**

<string> -- Định nghĩa lớp string

d. **Streams và Input/Output:**

<fstream> -- Định nghĩa các lớp hỗ trợ các phép toán cho dòng xuất nhập (*istream*) trên các chuỗi chứa trong các tập tin bên ngoài.

<ios> -- Định nghĩa nhiều kiểu và hàm cơ bản cho các xử lý của các dòng xuất nhập. Tiêu đề này thường bao gồm nhiều tiêu đề dòng xuất nhập khác mà ít khi được khai báo qua lệnh #include trực tiếp.

<iostream> -- Định nghĩa các đối tượng để đọc và viết ra các dòng tiêu chuẩn (*standard stream*). Đây thường là tiêu đề duy nhất cần phải bao gồm để tiến hành xuất và nhập (dữ liệu) từ C++.

<iosfwd> -- Khai báo trước các tham chiếu cho nhiều khuôn mẫu để dùng suốt quá trình xuất nhập dòng. Tất cả các khuôn mẫu như vậy được định nghĩa trong các tập tin tiêu đề chuẩn khác. Chỉ xử dụng tập tin tiêu đề này khi cần có một trong số các khai báo của nó, nhưng không phải là định nghĩa của nó.

<iomanip> -- Định nghĩa nhiều phép điều chỉnh mà nó lấy vào một đối số.

<istream> -- Định nghĩa lớp khuôn mẫu basic_istream, khuôn mẫu này làm trung gian cho các sự chèn vào và lấy ra (dữ liệu). Tiêu đề cũng định nghĩa một bộ điều chỉnh. Tập tin tiêu đề này thường được sử dụng bởi các tiêu đề về dòng xuất nhập khác, nó hiếm khi được bao gồm trực tiếp trong mã nguồn của người lập trình.

<ostream> -- Định nghĩa lớp khuôn mẫu basic_ostream, khuôn mẫu này làm trung gian cho các sự chèn (dữ liệu) vào cho các dòng xuất nhập. Tiêu đề cũng định nghĩa một bộ điều chỉnh. Tập tin tiêu đề này thường được sử dụng bởi các tiêu đề về dòng xuất nhập khác, nó hiếm khi được bao gồm trực tiếp trong mã nguồn của người lập trình.

<sstream> -- Định nghĩa nhiều lớp khuôn mẫu mà nó hỗ trợ các phép toán của dòng xuất nhập trên các chuỗi (dữ liệu) chứa trong một đối tượng mảng đã được cấp phát vùng nhớ. Những chuỗi này rất dễ chuyển đổi để tạo thành các đối tượng của lớp khuôn mẫu basic_string.

<streambuf> -- Bao gồm tiêu đề chuẩn cho các dòng xuất nhập của <streambuf> dùng để định nghĩa lớp khuôn mẫu basic_streambuf. Lớp này là cơ sở cho phép toán của các lớp của dòng xuất nhập. Tiêu đề này thường được bao gồm sẵn trong các khuôn mẫu về dòng xuất nhập khác, nó hiếm khi được bao gồm trực tiếp trong mã nguồn của người lập trình.

e. **Các số**

<complex> -- Định nghĩa lớp khuôn mẫu complex và hỗ trợ nhiều khuôn mẫu.

<numeric> -- Định nghĩa các hàm khuôn mẫu dùng chứa nhằm tiến hành các thuật toán cung cấp cho việc xử lý số.

<valarray> -- Định nghĩa lớp khuôn mẫu valarray hỗ trợ các lớp khuôn mẫu và các hàm.

f. **Hỗ trợ ngôn ngữ lập trình**

<exception> -- Định nghĩa nhiều kiểu và hàm liên hệ tới việc xử lý các ngoại lệ. Việc xử lý ngoại lệ được dùng trong tình huống mà máy tính có thể phục

hồi từ các lỗi. Nó cung ứng ý nghĩa cho việc trả về quyền điều khiển cho chương trình.

<limits> -- Định nghĩa các lớp khuôn mẫu numeric_limits và lo về việc biểu thị thứ tự của các điểm động và việc làm tròn.

<new> -- Định nghĩa nhiều kiểu và hàm mà nó điều khiển việc cấp phát cũng như trả về (bộ nhớ) của kho dự trữ dưới sự kiểm soát của chương trình. Nó cũng định nghĩa các thành phần cho việc báo cáo về quản lý của kho dự trữ.

<typeinfo> -- Hỗ trợ cho nhận dạng kiểu động.