

"Sơ đồ lớp"

Các mô hình tĩnh cung cấp sự biểu diễn cấu trúc của hệ thống dưới dạng các khối xây dựng cơ bản và mối quan hệ giữa chúng. "Bản chất tĩnh" của các mô hình này là chúng không thể hiện động lực của những thay đổi trong hệ thống theo thời gian. Đồng thời, cần hiểu rằng các mô hình này không chỉ chứa các mô tả cấu trúc mà còn chứa các mô tả về các hoạt động thực hiện hành vi nhất định của hệ thống. Phương tiện chính để biểu diễn các mô hình tĩnh là sơ đồ lớp. Các đỉnh của sơ đồ lớp được nạp các lớp và các cung (cạnh) được

mối quan hệ giữa chúng. Sơ đồ được sử dụng:

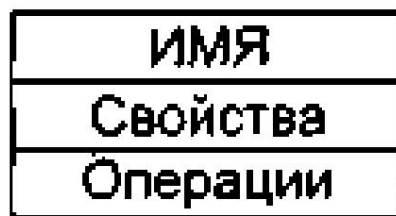
trong quá trình phân tích - để chỉ ra vai trò và trách nhiệm của các thực thể cung cấp hành vi của hệ thống;

trong quá trình thiết kế - để nắm bắt cấu trúc của các lớp tạo thành kiến trúc hệ thống.

Các nút trong sơ đồ lớp

Vì vậy, trên cùng trong sơ đồ lớp là lớp. Ký hiệu lớp được hiển thị trong Hình.

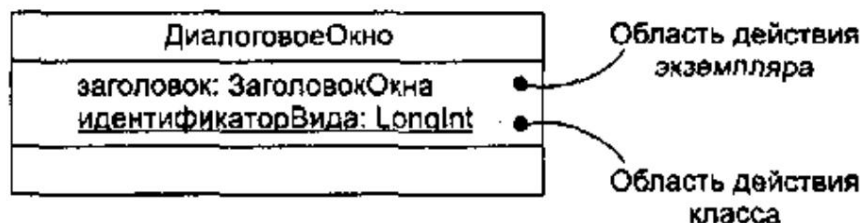
1.



См. 1. Chỉ định lớp

Tên lớp luôn được chỉ định, các thuộc tính và hoạt động được chỉ định một cách chọn lọc. Có thể thiết lập phạm vi của thuộc tính (hoạt động). Nếu một thuộc tính (thao tác) được gạch chân thì phạm vi của nó là lớp, nếu không thì phạm vi của Hành động là thể hiện (Hình 2).

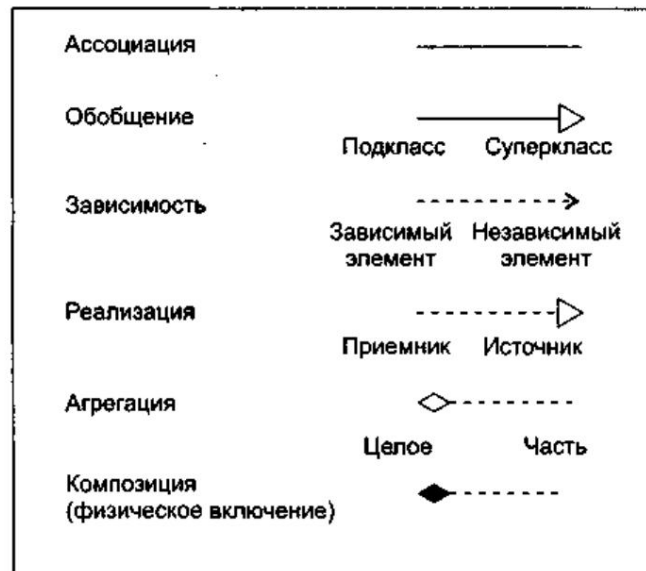
Nó có nghĩa là gì? Nếu phạm vi của một thuộc tính là một lớp thì tất cả các phiên bản (đối tượng) của nó có chung giá trị thuộc tính, nếu không thì mỗi phiên bản có giá trị thuộc tính riêng.



См. 2. Thuộc tính của lớp và cấp độ cá thể

Các mối quan hệ trong sơ đồ lớp

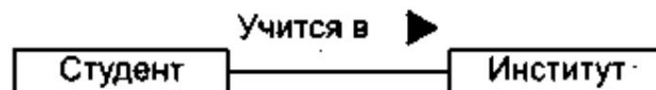
Các mối quan hệ được sử dụng trong sơ đồ lớp được thể hiện trong Hình 2. 5.



Cơ. 5. Các mối quan hệ trong sơ đồ lớp

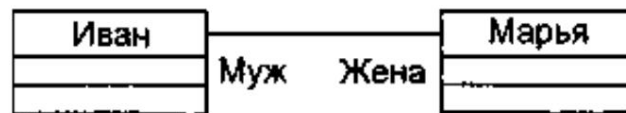
Các liên kết phản ánh mối quan hệ cấu trúc giữa các thể hiện của lớp, nghĩa là các kết nối giữa các đối tượng. Mỗi liên kết có thể có một nhãn- tên mô tả bản chất của mối quan hệ. Như thể hiện trong hình. 6, tên có thể được chỉ hướng - chỉ cần thêm một hình tam giác chỉ hướng, cho biết hướng đã cho

đọc tên.



Cơ. 6. Tên hiệp hội

Khi một lớp tham gia vào một hiệp hội, nó đóng một vai trò trong vấn đề này. Như thể hiện trong hình. 11.7, vai trò xác định cách lớp ở một đầu của liên kết xuất hiện với lớp ở đầu đối diện của liên kết.

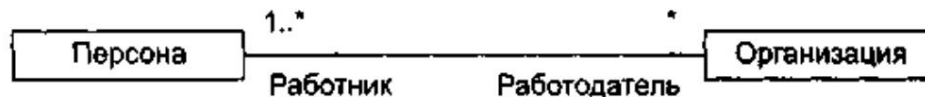


Cm. 7. Vai trò

Cùng một lớp trong các hiệp hội khác nhau có thể đóng các vai trò khác nhau. Điều quan trọng là phải biết có bao nhiêu đối tượng có thể được kết nối thông qua một thể hiện liên kết. Đại lượng này được gọi là độ sai lệch của vai trò trong liên kết và được viết dưới dạng biểu thức chỉ định một phạm vi giá trị hoặc một giá trị (Hình 8).

Bản ghi công suất ở một đầu của liên kết xác định số lượng đối tượng được kết nối với mỗi đối tượng ở đầu đối diện của liên kết. Ví dụ: bạn có thể đặt các tùy chọn nguồn sau:

- 5 - chính xác là năm;
- * - số lượng không giới hạn;
- 0..* - không hoặc nhiều hơn;
- 1..* - một hoặc nhiều;
- 3..7 - phạm vi nhất định;
- 1..3, 7 - một phạm vi hoặc số nhất định.



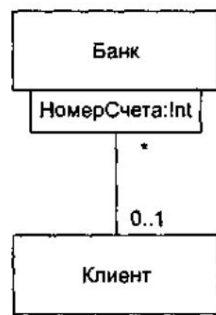
Cm. 8. Quyền lực

Vấn đề sau thường nảy sinh: làm thế nào một đối tượng ở một đầu của liên kết có thể được phân bổ một tập hợp các đối tượng ở đầu đối diện? Ví dụ: hãy xem xét sự tương tác giữa ngân hàng và khách hàng -

nhà đầu tư. Như thể hiện trong hình. 9, chúng ta thiết lập mối liên kết giữa lớp Bank và lớp Client. Trong bối cảnh Ngân hàng, chúng tôi có Số tài khoản, cho phép chúng tôi xác định một Khách hàng cụ thể. Theo nghĩa này, Số tài khoản là một thuộc tính của liên kết. Đó không phải là đặc điểm của Khách hàng vì Khách hàng không cần biết các thông số dịch vụ trong tài khoản của mình. Bây giờ, đối với một phiên bản nhất định của Ngân hàng và một giá trị nhất định của Số tài khoản, có thể xác định 0 hoặc một phiên bản của Khách hàng. Trong UML, để giải quyết vấn đề này, một vòng loại được giới thiệu - một thuộc tính liên kết, có các giá trị làm nổi bật một tập hợp các đối tượng được liên kết với đối tượng. Vòng loại được biểu thị bằng một hình chữ nhật gắn ở cuối liên kết. Thuộc tính-thuộc tính liên kết-khớp với hình chữ nhật.

sự kết hợp.

bé nhỏ



Cm. 9. Trình độ chuyên môn

Ngoài ra, vai trò trong liên kết có thể có cờ hiển thị. Ví dụ, trong hình. 10 cho thấy mối liên hệ giữa Ông Chủ và Người Phụ Nữ, cũng như giữa Người Phụ Nữ và Câu Đố. Đối với một phiên bản Chief nhất định, bạn có thể xác định các phiên bản Woman tương ứng. Mặt khác, Câu đố là riêng tư đối với Người phụ nữ nên không thể truy cập được từ bên ngoài. Như trong hình, bạn có thể di chuyển từ đối tượng Chief sang các thể hiện Woman (và ngược lại), nhưng bạn không thể nhìn thấy các thể hiện Riddle

dành cho đối tượng Phụ nữ.



Cm. 10. Tầm nhìn

Khi kết thúc liên kết, bạn có thể đặt ba cấp độ hiển thị bằng cách thêm biểu tượng hiển thị vào tên vai trò:

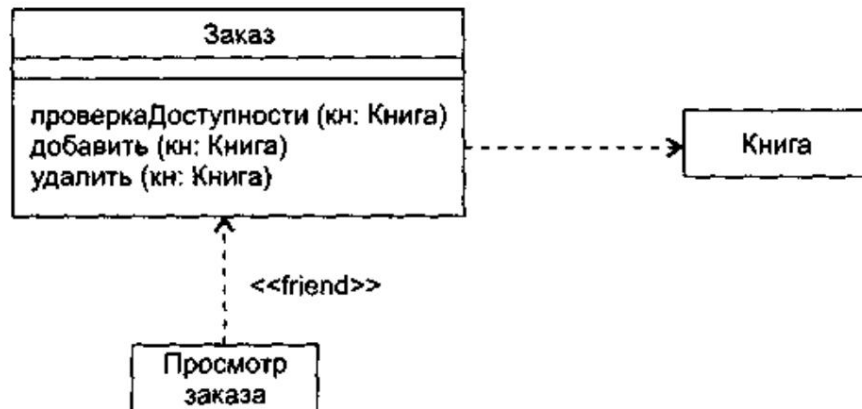
Theo mặc định, vai trò này được đặt ở chế độ hiển thị công khai;
khả năng hiển thị riêng tư chỉ ra rằng các đối tượng ở đầu này không
thể truy cập được đối với bất kỳ đối tượng nào bên ngoài liên kết;

khả năng hiển thị được bảo vệ (được bảo vệ) chỉ ra rằng các đối tượng ở đầu này
không thể truy cập được bởi bất kỳ đối tượng nào bên ngoài liên kết, ngoại trừ các con của lớp được
chỉ định ở đầu đối diện của liên kết.

Phần phụ thuộc là mối quan hệ sử dụng giữa máy khách (phần tử phụ thuộc) và nhà cung cấp (phần tử độc lập). Thông thường các hoạt động của khách hàng:

hoạt động của nhà cung cấp cuộc gọi;
có chữ ký trong đó giá trị trả về hoặc
các đối số thuộc về lớp nhà cung cấp.

Ví dụ, trong hình. Hình 12 cho thấy sự phụ thuộc của lớp Đơn hàng vào lớp Sách, vì Sách được sử dụng trong các hoạt động kiểm tra Tính sẵn có, thêm và xóa của lớp Đơn hàng.



Cơ. 12. Mối quan hệ phụ thuộc

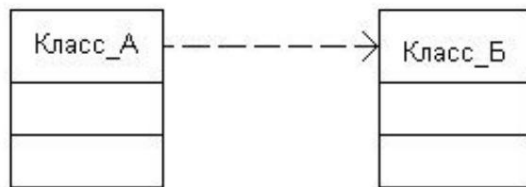
Hình này cho thấy một sự phụ thuộc khác cho thấy lớp Order View sử dụng lớp Order. Hơn nữa, Lệnh không hề biết gì về việc Xem Lệnh. Sự phụ thuộc này được gắn nhãn với khuôn mẫu "người bạn", mở rộng sự phụ thuộc đơn giản được xác định trong ngôn ngữ. Lưu ý rằng mối quan hệ phụ thuộc rất đa dạng - hiện tại ngôn ngữ cung cấp 17 loại phụ thuộc, được phân biệt bằng các khuôn mẫu.

Mối quan hệ phụ thuộc thường chỉ ra một số mối quan hệ ngữ nghĩa giữa hai phần tử của một mô hình hoặc hai tập hợp các phần tử đó, không phải là mối quan hệ liên kết, khái quát hóa hoặc triển khai. Nó chỉ liên quan đến các yếu tố của mô hình và không yêu cầu nhiều ví dụ riêng lẻ để giải thích ý nghĩa của nó. Mối quan hệ phụ thuộc được sử dụng trong trường hợp một số thay đổi trong một thành phần của mô hình có thể yêu cầu thay đổi ở thành phần khác.

phần tử mô hình phụ thuộc vào nó.

Mối quan hệ phụ thuộc được thể hiện bằng đồ họa bằng đường chấm chấm giữa các phần tử tương ứng với một mũi tên ở một đầu ("→" hoặc "←"). Trong sơ đồ lớp, mối quan hệ này kết nối các lớp riêng lẻ với nhau, với mũi tên hướng từ lớp khách của phần phụ thuộc đến lớp độc lập hoặc lớp nguồn (Hình 13). Hình này hiển thị hai lớp: Class_A và Class_B, trong khi Class_B là nguồn của một số sự phụ thuộc và Class_A là

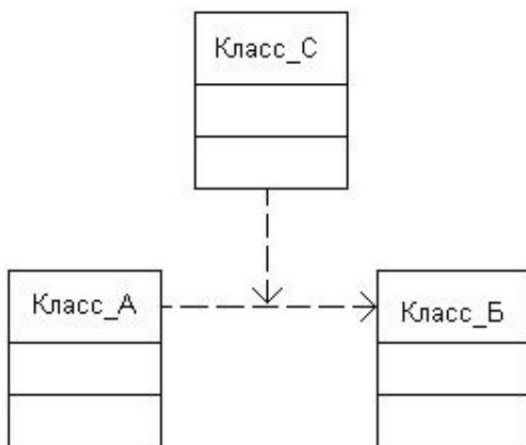
khách hàng của chứng nghiệm này.



См. 13. Biểu diễn bằng đồ họa mối quan hệ phụ thuộc trên

sơ đồ lớp Toàn bộ tập hợp

các phần tử mô hình có thể hoạt động như một lớp khách và một lớp nguồn phụ thuộc. Trong trường hợp này, một dòng mũi tên đến từ nguồn phụ thuộc sẽ chia thành nhiều dòng riêng biệt tại một số điểm, mỗi dòng có một mũi tên riêng cho lớp máy khách. Ví dụ: nếu chức năng của Class_C phụ thuộc vào các tính năng triển khai của Class_A và Class_B thì sự phụ thuộc này có thể được mô tả như sau (Hình 14).



См. 14. Biểu diễn đồ họa của sự phụ thuộc giữa lớp - máy khách (Class_C) và các lớp nguồn (Class_A và Class_B)

Mũi tên có thể được đánh dấu bằng từ khóa tùy chọn nhưng tiêu chuẩn trong dấu ngoặc kép và tên tùy chỉnh tùy chọn. Mỗi quan hệ phụ thuộc có các từ khóa được xác định trước biểu thị một số loại phụ thuộc đặc biệt nhất định. Những từ khóa này (khung mẫu) được viết trong dấu ngoặc kép bên cạnh mũi tên tương ứng với mỗi quan hệ này. Ví dụ về các khung mẫu cho các mối quan hệ phụ thuộc được trình bày dưới đây:

"quyền truy cập" - được sử dụng để biểu thị tính sẵn có của

•

thuộc tính và hoạt động của lớp nguồn cho lớp khách hàng;

- "liên kết" - lớp máy khách có thể sử dụng một số mẫu cho tham số hóa tiếp theo của nó; "dẫn xuất" - các thuộc tính
- của lớp khách hàng có thể được tính toán theo thuộc tính của lớp nguồn;
- "nhập" - các thuộc tính và hoạt động công khai của lớp nguồn trở thành một phần của lớp máy khách, như thể chúng được khai báo trực tiếp trong đó;

- "tinh chỉnh" - chỉ ra rằng lớp khách đóng vai trò sàng lọc lớp nguồn vì các lý do lịch sử khi có thêm thông tin trong quá trình thực hiện dự án.

Khái quát hóa là mối quan hệ giữa một chủ đề chung (lớp cha) và sự đa dạng chuyên biệt của chủ đề đó (lớp con). Một lớp con có thể có một lớp cha (một lớp cha) hoặc nhiều lớp cha (nhiều lớp cha). Trong trường hợp thứ hai chúng ta nói về số nhiều

sự kế thừa.

Như thể hiện trong hình. 15, lớp con Tủ Quần Áo Bay là sự kế thừa của siêu lớp Siêu Lớp Đồ Bay và Kho Đồ. Lớp con này kế thừa tất cả các thuộc tính và hoạt động của hai lớp cha của nó.

Đa kế thừa khá phức tạp, xảo quyệt và có nhiều cạm bẫy. Ví dụ: lớp con Apple_Pie không được bắt nguồn từ siêu lớp Pie và Apple. Đây là một cách sử dụng sai mục đích điển hình của đa kế thừa: một đứa trẻ kế thừa tất cả các thuộc tính từ cha mẹ của nó, mặc dù thông thường không phải tất cả các thuộc tính đều áp dụng cho đứa trẻ đó. Rõ ràng, Apple_Pie là Pie nhưng không phải là Apple, vì bánh nướng không mọc trên cây.



Cơ. 15. Đa kế thừa

Những vấn đề khó khăn hơn còn phát sinh khi kế thừa từ hai lớp có chung cha. Điều này được cho là dẫn đến một mạng lưới di truyền hình kim cương (Hình 16).



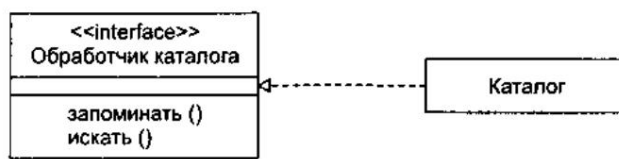
Cơ. 16 Mạng thừa kế hình kim cương

Chúng tôi tin rằng trong các lớp con Waiter và Singer, hoạt động của superclass Worker được định nghĩa lại phù hợp với nhiệm vụ

phân lớp (công việc của người phục vụ là phục vụ đồ ăn và công việc của ca sĩ là hát). Câu hỏi đặt ra: Singing_Waiter sẽ kế thừa phiên bản hoạt động nào? Làm gì với tài sản được thừa kế từ cha mẹ và tổ tiên chung? Chúng ta muốn có nhiều bản sao của một thuộc tính hay chỉ một bản sao?

Tất cả những vấn đề này làm tăng sự phức tạp của việc thực hiện và dẫn đến đưa ra nhiều quy tắc để xử lý các trường hợp đặc biệt.

Việc triển khai là mối quan hệ ngữ nghĩa giữa các lớp trong đó lớp đích thực hiện các hoạt động trên giao diện của lớp nguồn. Ví dụ, trong hình. Hình 17 cho thấy lớp Directory phải triển khai giao diện Directory Handler, nghĩa là Directory Handler được coi là nguồn và Directory là phần chìm.



Cơ. 17. Triển khai giao diện

Giao diện Người xử lý danh mục cho phép khách hàng tương tác với các đối tượng của lớp Thư mục mà không có kiến thức về kỷ luật truy cập được triển khai ở đây (LIFO - vào sau, ra trước; FIFO - vào trước, ra trước, v.v.).

Mối quan hệ khái quát hóa là mối quan hệ phân loại thông thường giữa phần tử tổng quát hơn (mẹ hoặc tổ tiên) và phần tử cụ thể hoặc chuyên biệt hơn (con hoặc hậu duệ). Mối quan hệ này có thể được sử dụng để thể hiện mối quan hệ giữa các gói, lớp, trường hợp sử dụng và các thành phần khác của ngôn ngữ UML. Khi được áp dụng cho sơ đồ lớp, mối quan hệ này mô tả cấu trúc phân cấp của các lớp và sự kế thừa các thuộc tính và hành vi của chúng. Điều này giả định rằng lớp con có tất cả các thuộc tính và hành vi của lớp tổ tiên,

đồng thời cũng có các thuộc tính và hành vi riêng mà lớp tổ tiên không có. Trong sơ đồ, mối quan hệ khái quát hóa được biểu thị bằng một đường liền nét có mũi tên hình tam giác ở một đầu (Hình 18). Một mũi tên biểu thị một lớp tổng quát hơn (lớp tổ tiên hoặc siêu lớp), trong khi sự vắng mặt của nó biểu thị một lớp chuyên biệt hơn (lớp con cháu hoặc lớp con).

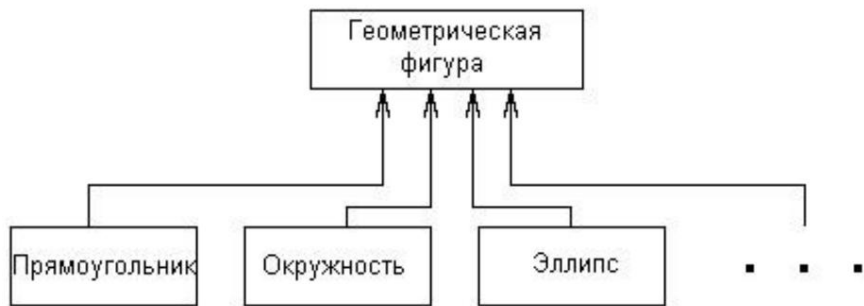


Cơ. 18. Biểu diễn đồ họa mối quan hệ khái quát hóa trong ngôn ngữ UML

Thông thường, một sơ đồ có thể hiển thị một số dòng cho một mối quan hệ khái quát, phản ánh bản chất phân loại của nó.

Trong trường hợp này, một lớp tổng quát hơn được chia thành các lớp con theo một quan hệ Tổng quát hóa. Ví dụ, lớp `Geometric_shape_on_plane`

(chữ in nghiêng biểu thị một lớp trừu tượng) có thể hoạt động như một siêu lớp cho các lớp con tương ứng với các hình dạng hình học cụ thể, chẳng hạn như Hình chữ nhật, Hình tròn, Hình elip, v.v. Thực tế này có thể được biểu diễn bằng đồ họa dưới dạng sơ đồ lớp của loại sau (Hình 19).



Hình 19. Một ví dụ về biểu diễn đồ họa của mối quan hệ khái quát hóa lớp học

Để đơn giản hóa ký hiệu trên sơ đồ lớp, một tập hợp các dòng biểu thị cùng một mối quan hệ tổng quát có thể được kết hợp thành một dòng. Trong trường hợp này, những đường riêng lẻ này được mô tả hội tụ trên một mũi tên duy nhất có một điểm giao nhau với chúng (Hình 20).



Cơ. 20. Một biến thể của biểu diễn đồ họa của mối quan hệ tổng quát hóa lớp trong trường hợp kết hợp các dòng riêng lẻ

Ký hiệu này về mặt hình thức tương ứng với một loại biểu đồ đặc biệt đã được thảo luận trong Chương 2, cụ thể là cây phân cấp. Trong trường hợp này, lớp tổ tiên là gốc của cây này và các lớp con cháu là lá của nó. Sự khác biệt là sơ đồ lớp có thể chỉ ra tiềm năng của các lớp con cháu khác không có trong biểu tượng sơ đồ lớp (hình elip thay vì hình chữ nhật).

Bên cạnh mũi tên khái quát hóa, có thể đặt một dòng văn bản chỉ ra một số thuộc tính bổ sung của mối quan hệ này. Văn bản này sẽ áp dụng cho tất cả các dòng khái quát hóa đi đến các lớp con cháu. Nói cách khác, thuộc tính được ghi chú áp dụng cho tất cả các lớp con của một quan hệ nhất định. Trong trường hợp này, văn bản phải được coi là một hạn chế và sau đó nó được viết trong dấu ngoặc nhọn.

Các mối quan hệ tổng hợp và thành phần trong UML được coi là các kiểu liên kết, được sử dụng để thể hiện các mối quan hệ cấu trúc giữa một "toàn bộ" (tổng hợp) và các "bộ phận" của nó. Tập hợp thể hiện mối quan hệ bằng tham chiếu (chỉ các con trỏ tới các phần được bao gồm trong tổng hợp), thành phần hiển thị mối quan hệ bao gồm vật lý (bản thân các phần đó được bao gồm trong tổng hợp).

Một mối quan hệ tổng hợp xảy ra giữa một số lớp nếu một trong các lớp đại diện cho một thực thể bao gồm các thực thể khác làm thành phần.

Mối quan hệ này có tầm quan trọng cơ bản để mô tả cấu trúc của các hệ thống phức tạp, vì nó được sử dụng để biểu diễn các mối quan hệ hệ thống thuộc loại "bộ phận-toàn bộ". Tiết lộ cấu trúc bên trong của hệ thống, mối quan hệ tổng hợp cho thấy hệ thống bao gồm những thành phần nào và chúng liên quan với nhau như thế nào. Từ quan điểm của mô hình, các bộ phận riêng lẻ của hệ thống có thể hoạt động cả dưới dạng các phần tử và dưới dạng hệ thống con, do đó, cũng có thể tạo thành các thành phần tổng hợp hoặc hệ thống con. Mối quan hệ này vốn mô tả sự phân rã hoặc chia nhỏ một hệ thống phức tạp thành các phần thành phần đơn giản hơn, những phần này cũng có thể được phân tách nếu có nhu cầu sau này.

Về mặt đồ họa, mối quan hệ tổng hợp được mô tả bằng một đường liền nét, một trong hai đầu của nó là một viên kim cương chưa được lấp đầy bên trong. Viên kim cương này cho biết lớp nào đại diện cho "tổng thể". Các lớp còn lại là "các bộ phận" của nó (Hình 22).



См. 22. Biểu diễn đồ họa của quan hệ tập hợp trong ngôn ngữ UML Một ví dụ khác về quan hệ tập hợp có thể là việc chia máy tính cá nhân thành các bộ phận cấu thành của nó mà mỗi độc giả đều biết: đơn vị hệ thống, màn hình, bàn phím và chuột. Bằng cách sử dụng các ký hiệu ngôn ngữ UML, thành phần thành phần của PC có thể được biểu diễn dưới dạng sơ đồ lớp tương ứng (Hình 23), trong trường hợp này minh họa mối quan hệ tổng hợp.



См. 23. Sơ đồ lớp để minh họa mối quan hệ tổng hợp sử dụng PC làm ví dụ

Quan hệ tổng hợp Quan hệ tổng hợp,

như đã đề cập trước đó, là một trường hợp đặc biệt của quan hệ tổng hợp. Mỗi quan hệ này nhằm làm nổi bật một dạng đặc biệt của mối quan hệ "bộ phận-toàn thể", trong đó các bộ phận cấu thành, theo một nghĩa nào đó, nằm trong tổng thể. Tính đặc thù của mối quan hệ giữa chúng nằm ở chỗ các bộ phận không thể hoạt động tách biệt với tổng thể, tức là khi tổng thể bị phá hủy thì tất cả các bộ phận cấu thành của nó cũng bị phá hủy. Có lẽ không phải là tốt nhất nhưng chắc chắn là một ví dụ rõ ràng về mối quan hệ này đối với mọi người, đó là tế bào sống trong sinh học. Một ví dụ khác là cửa sổ giao diện chương trình, có thể

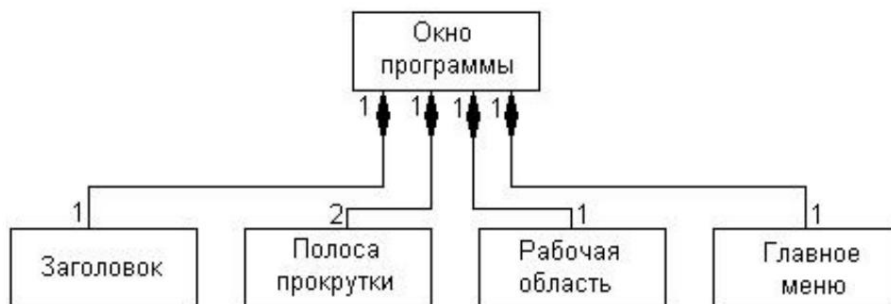
bao gồm thanh tiêu đề, nút điều khiển kích thước, thanh cuộn, menu chính, vùng làm việc và thanh trạng thái. Để hiểu rằng một cửa sổ như vậy là một lớp, các thành phần của nó vừa là lớp, vừa là thuộc tính hoặc thuộc tính của cửa sổ. Tình huống cuối cùng rất đặc trưng của mối quan hệ thành phần, vì nó phản ánh những cách khác nhau để thể hiện mối quan hệ này.

Về mặt đồ họa, mối quan hệ của bố cục được mô tả bằng một đường liền nét, một trong hai đầu của nó là một viên kim cương được tô bóng bên trong. Hình thoi này cho biết lớp nào là lớp thành phần hoặc "toàn bộ". Các lớp còn lại là "các bộ phận" của nó (Hình 24).



См. 24 Biểu diễn đồ họa của mối quan hệ thành phần trong UML

Các ký hiệu bổ sung cho mối quan hệ thành phần và tập hợp có thể được sử dụng làm ký hiệu bổ sung cho mối quan hệ liên kết. Cụ thể, chỉ ra tính đa dạng của lớp liên kết và tên của liên kết này, là tùy chọn. Liên quan đến ví dụ về lớp "Program_Window" được mô tả ở trên, sơ đồ lớp của nó có thể trông như thế này (Hình 5.11).

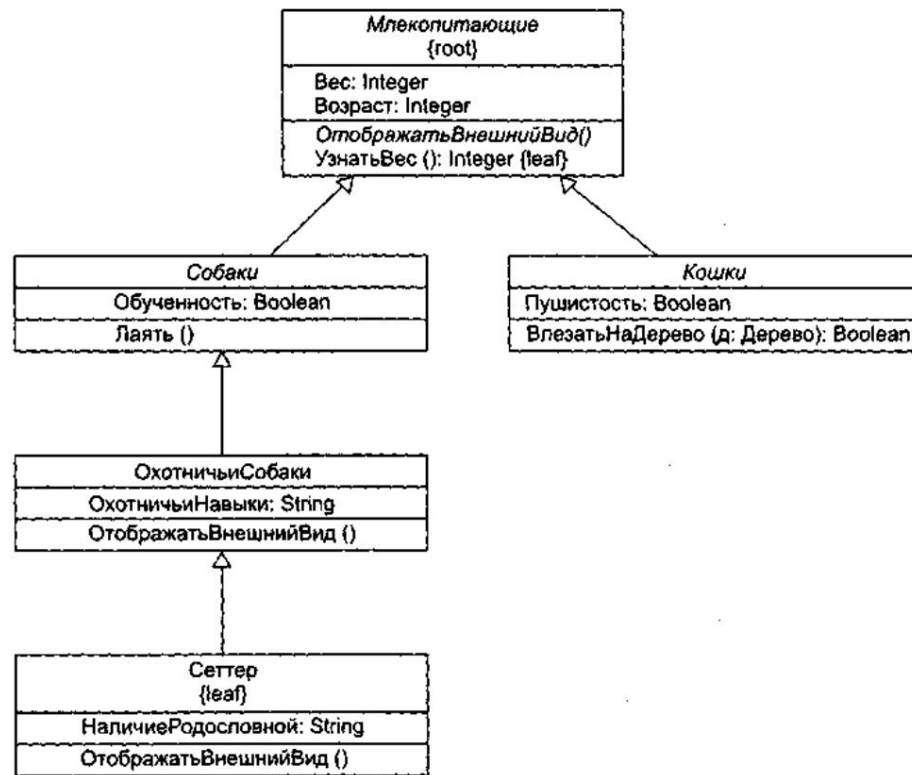


См. 25. Sơ đồ lớp để minh họa mối quan hệ thành phần bằng cách sử dụng ví dụ về lớp cửa sổ chương trình. Ví dụ này có thể minh họa các tính năng khác

của chương trình máy tính đang được phát triển mà không được chỉ ra rõ ràng khi mô tả ví dụ này. Vì vậy, cụ thể, việc chỉ ra bội số của 1 bên cạnh lớp "Work_area" là điển hình cho các ứng dụng một tài liệu.

Cây kế thừa

Khi sử dụng quan hệ khái quát hóa, hệ thống phân cấp lớp được xây dựng. Một số lớp trong hệ thống phân cấp này có thể trừu tượng. Một lớp trừu tượng là một lớp không thể có các thể hiện. Tên lớp trừu tượng được viết in nghiêng. Ví dụ, trong hình. 26 hiển thị các lớp trừu tượng Động vật có vú, Chó, Mèo.



См. 26. Tính trừu tượng và đa hình

Ngoài ra, còn có các lớp Chó sẵn cụ thể, Setter, mỗi cái có thể có phiên bản.

Thông thường, một lớp kế thừa một số đặc điểm của lớp cha và truyền các đặc điểm đó cho lớp con của nó. Đôi khi bạn cần xác định một lớp cuối cùng không thể có con. Các lớp như vậy được đánh dấu bằng lá giá trị thẻ (đặc trưng), được viết sau tên lớp. Ví dụ: hình minh họa lớp Setter cuối cùng.

Đôi khi việc đánh dấu một lớp gốc không thể có lớp cha là rất hữu ích. Một lớp như vậy được đánh dấu bằng gốc giá trị thẻ (đặc trưng), được viết sau tên lớp. Ví dụ, hình vẽ thể hiện lớp gốc Động vật có vú.

Hoạt động có tính chất tương tự. Thông thường, một thao tác có tính đa hình, nghĩa là các thao tác có chữ ký tương tự có thể được xác định tại các điểm khác nhau trong hệ thống phân cấp. Các hoạt động như vậy từ các lớp con sẽ ghi đè hành vi của các hoạt động tương ứng từ các lớp cha. Khi xử lý một tin nhắn (trong thời gian chạy), một lựa chọn đa hình được thực hiện bằng một trong các hoạt động phân cấp phù hợp với loại đối tượng. Ví dụ: DisplayAppearance() và

ClimbUpTree (sồi) - hoạt động đa hình. Ngoài ra, thao tác Mammals::DisplayAppearance () là trừu tượng, nghĩa là không đầy đủ và cần có phần tử con để thực hiện. Tên của một thao tác trừu tượng được viết bằng chữ in nghiêng (giống như tên lớp). Mặt khác, Mammals::FindWeight() là một phép toán hữu hạn, được biểu thị bằng đặc tính lá. Điều này có nghĩa là hoạt động không đa hình và không thể chồng chéo.

Giao diện Giao diện

là các thành phần của một sơ đồ biến thể. Tuy nhiên, khi xây dựng sơ đồ lớp, các giao diện riêng lẻ có thể được chỉ định, trong trường hợp đó, một ký hiệu đồ họa đặc biệt được sử dụng để mô tả chúng - một hình chữ nhật lớp với từ khóa hoặc “giao diện” khuôn mẫu (Hình 27). Trong trường hợp này, hình chữ nhật không có phần thuộc tính và chỉ có phần thao tác được chỉ định.



См. 27. Một ví dụ về biểu diễn đồ họa của giao diện trong sơ đồ lớp học