Александр Леоненков

UML

2-е издание

САМОУЧИТЕЛЬ

Объектно-ориентированный анализ

Проектирование программных систем

Диаграммы логического и физического моделирования

Документирование проекта в нотации UML

UML в Rational Rose 2002

Язык объектных ограничений OCL

Освой UML самостоятельно

УДК 681.3.068+800.92UML
ББК 32.973.26-018.1
Л47

Леоненков А. В.

Л47 Самоучитель UML. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2004. — 432 с.: ил.

ISBN 5-94157-342-1

Цель книги — помочь менеджерам и руководителям проектов, руководителям информационных служб, бизнес-аналитикам, корпоративным программистам и ведущим разработчикам самостоятельно освоить базовые концепции и понятия наиболее перспективной и современной методологии разработки корпоративных информационных систем для последующего применения полученных знаний в ходе выполнения реальных проектов и совершенствования бизнес-процессов с использованием соответствующих CASE-средств. Рассматриваются основы современной технологии унифицированного анализа и проектирования программных систем на языке UML. Подробно излагаются базовые понятия языка UML, необходимые для построения объектно-ориентированных моделей корпоративных программных систем с использованием специальной графической нотации. Приводятся конкретные рекомендации по изображению канонических диаграмм UML и рассматриваются особенности разработки моделей с помощью CASE-средства IBM Rational Rose 2002. Описывается нотация OCL — языка объектных ограничений, что делает книгу уникальной среди аналогичных изданий.

Для программистов

УДК 681.3.068+800.92UML
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. гл. редактора	*Игорь Шишигин*
Зав. редакцией	*Григорий Добин*
Редактор	*Андрей Смышляев*
Компьютерная верстка	*Натальи Караваевой*
Корректор	*Виктория Пиотровская*
Дизайн обложки	*Игоря Цырульникова*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 27.02.04.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 46,44.
Тираж 3 000 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

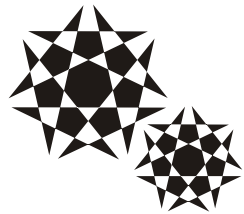
Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02
от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-342-1

Содержание

Глава 2



Исторический обзор развития методологии объектно-ориентированного анализа и проектирования сложных систем

2.1. Предыстория. Математические основы

Представление знаний о различных объектах и системах окружающего нас мира при помощи графической символики уходит своими истоками в глубокую древность. В качестве примеров можно привести условные обозначения знаков Зодиака, магические символы различных оккультных доктрин, графические изображения геометрических фигур на плоскости и в пространстве. Важным достоинством той или иной графической нотации является возможность образного закрепления содержательного смысла или семантики отдельных понятий, что существенно упрощает процесс общения между посвященными в соответствующие теории и идеологии.

2.1.1. Теория множеств

Как одну из наиболее известных систем графических символов, оказавших непосредственное влияние на развитие научного мышления, следует отметить язык диаграмм английского логика Джона Венна (1834—1923). В настоящее время *диаграммы Венна* применяются для иллюстрации основных теоретико-множественных операций, которые являются предметом специального раздела математики — *теории множеств*. Поскольку многие общие идеи моделирования систем имеют адекватное описание в терминологии теории множеств, рассмотрим основные понятия данной теории, имеющие отношение к современным концепциям и технологиям исследования сложных систем.

Исходным понятием теории множеств является само понятие *множество*, под которым принято понимать некоторую совокупность объектов, хорошо различимых нашей мыслью или интуицией. При этом не делается никаких

предположений ни о природе этих объектов, ни о способе их включения в данную совокупность. Отдельные объекты, составляющие то или иное множество, называют *элементами* данного множества. Вопрос "Почему мы рассматриваем ту или иную совокупность элементов как множество?" не требует ответа, поскольку в общее определение множества не входят никакие дополнительные условия на включение отдельных элементов в множество. Если нам хочется, например, рассмотреть множество, состоящее из трех элементов: солнце, море, апельсин, то никто не сможет запретить это сделать.

Примерами множеств являются: множество квартир жилого дома, множество книг в библиотеке, множество натуральных чисел, совокупность компьютеров в офисе, штат сотрудников фирмы, множество живущих на планете людей, множество звезд на небосводе.

Примечание

Создается впечатление, что ситуация с заданием множеств более или менее очевидна. Но это впечатление обманчиво. Даже не говоря об известных парадоксах теории множеств, как быть с множеством мыслей отдельного человека? Или множеством всех красок, которые встречаются в природе? Однако такие каверзные случаи мы рассматривать не будем, ограничив круг ситуаций такими, в которых идентификация отдельных элементов множеств не превращается в серьезную проблему. С другой стороны, процесс моделирования сложных систем зачастую сопряжен именно с подобного рода трудностями.

В теории множеств используется специальное соглашение, по которому множества обозначаются прописными буквами латинского алфавита, и традиция эта настолько общепризнанна, что не возникает никакого сомнения в ее целесообразности. При этом отдельные элементы обозначаются строчными буквами, иногда с индексами, которые вносят некоторую упорядоченность в последовательность рассмотрения этих элементов. Важно понимать, что какой бы то ни было порядок, вообще говоря, не входит в исходное определение множества.

Таким образом, множество, например, квартир 100-квартирного жилого дома с использованием специальных обозначений можно записать следующим образом: $A = \{a_1, a_2, a_3, ..., a_{100}\}$. Здесь фигурные скобки служат обозначением совокупности элементов, каждый из которых имеет свой уникальный числовой индекс. Важно понимать, что для данного конкретного множества элемент a_{10} обозначает отдельную квартиру в рассматриваемом жилом доме. При этом вовсе не обязательно, чтобы номер этой квартиры был равен 10, хотя с точки зрения удобства это было бы желательно.

Примечание

При выборе обозначений для множеств допускается некоторый произвол, который не всегда понятен лицам, далеким от математики. Однако здесь уместна

аналогия с выбором имен для переменных и процедур в языках высокого уровня, когда программист сам решает, как ему обозначать соответствующую конструкцию в программе.

Принято называть элементы отдельного множества *принадлежащими* данному множеству. Данный факт записывается при помощи специального символа "$\in$", который так и называется — *символ принадлежности*. Например, запись $a_{10} \in A$ означает тот простой факт, что отдельная квартира (возможно, с номером 10) принадлежит рассматриваемому множеству квартир некоторого жилого дома.

Следующим важным понятием, которое служит прототипом многих более конкретных терминов при моделировании сложных систем, является понятие *подмножества*. Казалось бы, интуитивно и здесь нет ничего неясного. Если есть некоторая совокупность, рассматриваемая как множество, то любая ее часть и будет являться подмножеством. Так, например, совокупность квартир на первом этаже жилого дома есть не что иное, как подмножество рассматриваемого нами примера. Ситуация становится не столь тривиальной, если рассматривать множество абстрактных понятий, таких как сущность или класс.

Для обозначения подмножества используется специальный символ. Если утверждается, что множество A является подмножеством множества B, то это записывается как $A \subset B$. Запоминать подобные значки не всегда удобно, поэтому со временем была предложена специальная система графических обозначений.

Как же используются диаграммы Венна в теории множеств? Тот факт, что некоторая совокупность элементов образует множество, можно обозначить графически в виде круга. В этом случае окружность имеет содержательный смысл или, выражаясь более точным языком, семантику *границы* данного множества. Тогда вполне логично отношение включения элементов одного множества в другое изобразить графически следующим образом (рис. 2.1). На этом рисунке большему множеству B соответствует внешний круг, а меньшему множеству (подмножеству) A — внутренний.

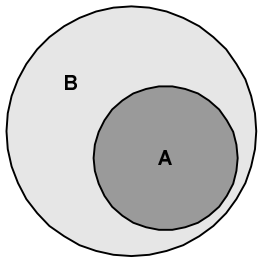


Рис. 2.1. Диаграмма Венна для отношения включения двух множеств

Подобным образом можно изобразить и основные теоретико-множественные операции. Так, в общем случае, *пересечением* двух множеств *A* и *B* называется некоторое третье множество *C*, которое состоит из тех элементов двух исходных множеств, которые *одновременно* принадлежат и множеству *A*, и множеству *B*. Для этой операции также имеется специальное обозначение: $C = A \cap B$. Например, если для операции пересечения в качестве множества *A* рассмотреть множество сотрудников некоторой фирмы, а в качестве множества *B* — множество всех мужчин, то нетрудно догадаться, что множество *C* будет состоять из элементов — всех сотрудников мужского пола данной фирмы. Операция пересечения множеств также может быть проиллюстрирована с помощью диаграмм Венна (рис. 2.2). На этом рисунке условно изображены два множества *A* и *B*, а затененной области как раз и соответствует множество *C*, являющееся пересечением множеств *A* и *B*.

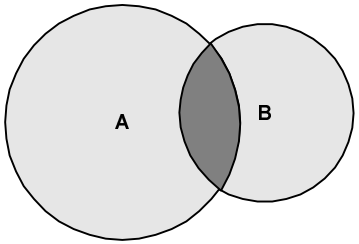


Рис. 2.2. Диаграмма Венна для пересечения двух множеств

Следующей операцией, которая также допускает наглядную интерпретацию, является *объединение* множеств. Под объединением двух множеств *A* и *B* понимается некоторое третье, пусть это будет множество *D*, которое состоит из тех элементов, которые принадлежат или *A*, или *B*, или им обоим одновременно. Конечно, специальное обозначение есть и для этой операции: $D = A \cup B$. Так, если в качестве *A* рассмотреть множество, состоящее из клавиатуры и мыши, а в качестве *B* — множество, состоящее из системного блока и монитора, то нетрудно догадаться, что их объединение, т. е. *D*, образует основные составляющие персонального компьютера. Эта операция также может быть условно изображена графически (рис. 2.3). На этом рисунке объединению соответствует затемненная область, только размеры и форма ее отличаются от случая пересечения на предыдущем рисунке.

Хотя существуют и другие операции над множествами, а также целый ряд с ними связанных дополнительных понятий, их рассмотрение выходит за рамки настоящей книги. Последнее, на что следовало бы обратить внимание при столь кратком знакомстве с основами теории множеств — это на так называемые понятия *мощности* множества и *отношения* множеств. Что касается понятия мощности, то данный термин важен для анализа кратности связей, поскольку ассоциируется с количеством элементов отдельного мно-

жества. В случае конечного множества ситуация очень простая, поскольку мощность конечного множества равна количеству его элементов. Таким образом, возвращаясь к примеру с множеством A квартир жилого дома, можно сказать, что его мощность равна 100.

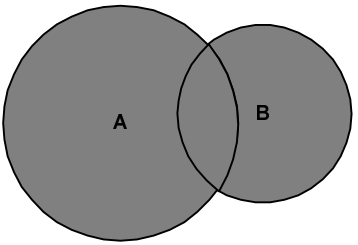


Рис. 2.3. Диаграмма Венна для объединения двух множеств

Ситуация усложняется, когда рассматриваются бесконечные множества. Не вдаваясь в технические детали, которые послужили источником драматичного по своим последствиям кризиса основ математики, ограничим наше рассмотрение бесконечными множествами *счетной* мощности. Ими принято считать множества, содержащие бесконечное число элементов, которые, однако, можно перенумеровать натуральными числами 1, 2, 3 и т. д. При этом важно иметь в виду, что достичь последнего элемента при такой нумерации принципиально невозможно, иначе множество окажется конечным.

Например, есть все основания считать множество всех звезд бесконечным, хотя многие из них имеют свое уникальное название. С другой стороны, множество всех возможных комбинаций из 8 символов, которые могут служить для ввода некоторого пароля, конечное, хотя и достаточно большое. Точнее, это множество имеет конечную мощность.

Примечание

Проблема бесконечного могла бы показаться отвлеченной и имеющей некоторый философский оттенок, если бы не ее связь с моделированием сложных систем. Так, при рассмотрении некоторой предметной области с целью построения ее модели приходится выделять конечное число сущностей, образующих определенный "скелет" или остов архитектуры будущей модели. И это при том, что реальность предметов допускает бесконечное рассмотрение их свойств, атрибутов и взаимосвязей.

Наконец, было упомянуто и следующее понятие, различные аспекты которого будут служить темой рассмотрения во всех последующих главах. Это фундаментальное понятие *отношения* множеств, которое часто заменяется терминами *связь* или *соотношение*. Этот термин ведет свое происхождение от теории множеств и служит для обозначения любого подмножества упорядоченных кортежей, построенных из элементов некоторых исходных

множеств. При этом под *кортежем* понимается просто набор или список элементов, важно только, чтобы они были упорядочены. Другими словами, если рассматривать первый элемент кортежа, то он всегда будет первым в списке элементов, второй элемент кортежа будет вторым элементом в списке и т. д. Можно ли это записать с использованием специальных обозначений?

Хотя и существует некоторая неоднозначность в принятых обозначениях, кортеж из двух элементов удобно обозначать как $<a_1, a_2>$, из трех — $<a_1, a_2, a_3>$ и т. д. При этом отдельные элементы могут принадлежать как одному и тому же, так и различным множествам. Важно иметь в виду, что порядок выбора элементов для построения кортежей строго фиксирован для конкретной задачи. Речь идет о том, что первый элемент всегда выбирается из первого множества, второй — из второго и т. д. Отношение в этом случае будет характеризовать способ или семантику выбора отдельных элементов из одного или нескольких множеств для подобного упорядоченного списка. В этом смысле взаимосвязь является частным случаем отношения, о чем будет сказано в последующем.

К сожалению, диаграммы Венна не предназначены для иллюстрации отношений в общем случае. Однако отношения послужили исходной идеей для развития другой теории, которая даже в своем названии несет отпечаток графической нотации, а именно — теории графов. В этой связи наиболее важным является тот факт, что теоретико-множественные отношения послужили также основой для разработки реляционной алгебры в теории реляционных баз данных. Развитие последней привело к тому, что в последние годы именно реляционные СУБД конкретных фирм доминируют на рынке соответствующего программного обеспечения.

2.1.2. Теория графов

Граф можно рассматривать как графическую нотацию для бинарного отношения двух множеств. Бинарное отношение состоит из таких кортежей или списков элементов, которые содержат только два элемента некоторого множества. Хотя основные понятия теории графов получили свое развитие задолго до появления теории множеств как самостоятельной научной дисциплины, формальное определение графа удобно представить в теоретико-множественных терминах.

Графом называется совокупность двух множеств: множества точек (или *вершин*) и множества соединяющих их линий (или *ребер*). Формально граф задается в виде двух множеств: $G = (V, E)$, где $V = \{v_1, v_2, ..., v_n\}$ — множество вершин графа, а $E = \{e_1, e_2, ..., e_m\}$ — множество ребер графа. Натуральное число n определяет общее количество вершин конкретного графа, а натуральное число m — общее количество ребер графа. Следует заметить, что в общем случае не все вершины графа могут соединяться между собой, что ставит в соответствие каждому графу некоторое бинарное отношение P_G,

состоящее из всех пар вида $<v_i, v_j>$, где $v_i, v_j \in V$. При этом пара $<v_i, v_j>$ и, соответственно, пара $<v_j, v_i>$ принадлежат отношению P_G в том случае, если вершины v_i и v_j соединяются в графе G некоторым ребром $e_k \in E$. Вершины графа изображаются точками, а ребра — отрезками прямых линий. Рядом с вершинами и ребрами записываются соответствующие номера (или идентификаторы), позволяющие идентифицировать их однозначным образом.

Вообще говоря, графы бывают двух различных типов. Рассмотренное выше определение относится к *неориентированному* графу, т. е. к такому графу, у которого связывающие вершины ребра не имеют направления или ориентации. Кроме неориентированных графов существуют *ориентированные* графы, которые определяются следующим образом.

Ориентированный граф также задается в виде двух множеств $G = (V, E)$, где $V = \{v_1, v_2, ..., v_n\}$ — множество вершин графа, а $E = \{e_1, e_2, ..., e_m\}$ — множество *дуг* графа. Натуральное число n определяет общее количество вершин конкретного графа, а натуральное число m — общее количество дуг графа. При этом каждая дуга $e_k \in E$ ориентированного графа G имеет свое начало — некоторую единственную вершину $v_i \in V$ и конец — некоторую единственную вершину $v_j \in V$. В отличие от ребра, дуга всегда имеет обозначение со стрелочкой, которая направлена к конечной вершине дуги. Множество дуг ставит в соответствие каждому ориентированному графу некоторое бинарное отношение P_G, состоящее из всех пар вида $<v_i, v_j>$, где $v_i, v_j \in V$. При этом пара $<v_i, v_j>$ принадлежит отношению P_G в том случае, если вершины v_i и v_j соединяются в графе G некоторой дугой $e_k \in E$ с началом в вершине v_i и концом в вершине v_j.

Ниже представлены два примера конкретных графов (рис. 2.4). При этом первый из них (рис. 2.4, *а*) является неориентированным графом, а второй (рис. 2.4, *б*) — ориентированным графом. Как нетрудно заметить, для неориентированного графа ребро e_1 соединяет вершины v_1 и v_2, ребро e_2 соединяет вершины v_1 и v_3, ребро e_3 соединяет вершины v_2 и v_3 и так далее. Последнее ребро, e_8, соединяет вершины v_4 и v_5, тем самым задается описание графа в целом. Других ребер данный граф не содержит, также как не содержит других вершин, не изображенных на рисунке. Так, хотя ребра e_6 и e_7 визуально пересекаются, но точка их пересечения не является вершиной графа.

Для ориентированного графа (рис. 2.4, *б*) ситуация несколько иная. А именно, вершины v_1 и v_2 соединены дугой e_1, для которой вершина v_2 является началом дуги, а вершина v_1 — концом этой дуги. Далее дуга e_2 соединяет вершины v_1 и v_4, при этом началом дуги e_2 является вершина v_1, а концом — вершина v_4.

Графы широко применяются для представления различной информации о структуре систем и процессов. Примерами подобных графических моделей

могут служить: схемы автомобильных дорог, соединяющих отдельные населенные пункты; схемы телекоммуникаций, используемых для передачи информации между отдельными узлами; схемы программ, на которых указываются варианты ветвления вычислительного процесса. Общим свойством подобных моделей является возможность представления информации в графическом виде в форме соответствующего графа. При этом отдельные модели, как правило, обладают дополнительной семантикой и специальными обозначениями, характерными для той или иной предметной области.

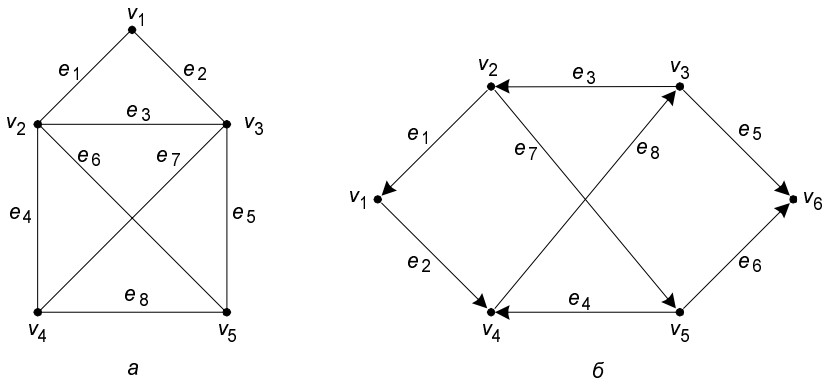


Рис. 2.4. Примеры неориентированного (*а*) и ориентированного (*б*) графов

Важными понятиями теории графов являются понятия маршрута и пути, которые ассоциируются с последовательным перемещением от вершины к вершине по соединяющим их ребрам или дугам. Для неориентированного графа *маршрут* определяется как конечная или бесконечная упорядоченная последовательность ребер $S = <..., e_{s1}, e_{s2}, ..., e_{sk},>$, таких, что каждые два соседних ребра имеют общую вершину. Нас будут интересовать только конечные маршруты $S = <e_{s1}, e_{s2}, ..., e_{sk}>$, т. е. такие, которые состоят из конечного числа ребер. При этом ребро e_{s1} принято считать началом маршрута S, а ребро e_{sk} — концом маршрута S. Для ориентированного графа соответствующая последовательность дуг $S = <e_{s1}, e_{s2}, ..., e_{sk}>$ называется *ориентированным маршрутом*, если две соседние дуги имеют общую вершину, которая является концом предыдущей и началом последующей дуги.

Примерами маршрутов для неориентированного графа (рис. 2.4, *а*) являются последовательности ребер: $S_1 = <e_1, e_2, e_5, e_8>$, $S_2 = <e_1, e_2, e_3, e_1>$, $S_3 = <e_3, e_5, e_8>$. Если в маршруте не повторяются ни ребра, ни вершины, как в случае S_1 и S_3, то такой неориентированный маршрут называется *простой цепью.*

Примерами ориентированных маршрутов для графа (рис. 2.4, *б*) являются такие последовательности дуг: $S_1 = <e_2, e_8, e_5>$, $S_2 = <e_3, e_7, e_6>$,

$S_3 = <e_8, e_3, e_7, e_4, e_8>$. Если в ориентированном маршруте не повторяются ни ребра, ни вершины, как в случае S_1 и S_2, то такой ориентированный маршрут называется *путем*. Последнее понятие также иногда применяется для обозначения простой цепи в неориентированных графах и для определения специального класса графов, так называемых деревьев. В общем случае деревья служат для графического представления иерархических структур или иерархий, занимающих важное место в ООАП.

Деревом в теории графов называется такой граф $D = <V, E>$, между любыми двумя вершинами которого существует единственная простая цепь, т. е. неориентированный маршрут, у которого вершины и ребра не повторяются. Применительно к ориентированным графам, соответствующее определение является более сложным, поскольку основывается на выделении некоторой специальной вершины v_0, которая получила специальное название *корневой* вершины или просто — *корня*. В этом случае ориентированный граф $D = <V, E>$ называется ориентированным деревом или сокращенно — деревом, если между корнем дерева v_0 и любой другой вершиной существует единственный путь, берущий начало в v_0. Ниже представлены два примера деревьев: неориентированного (рис. 2.5, *а*) и ориентированного (рис. 2.5, *б*).

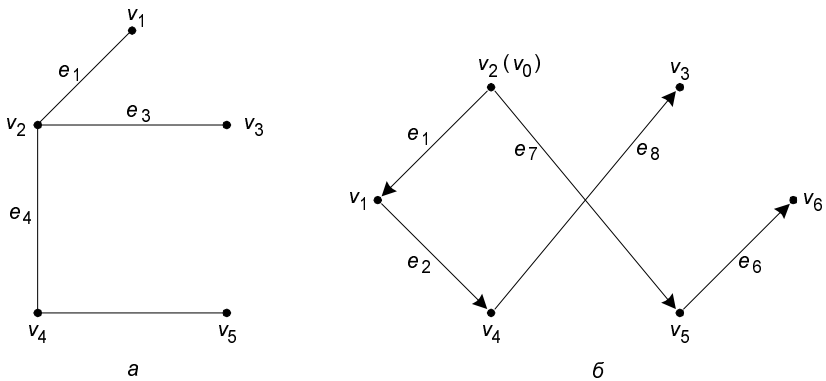


Рис. 2.5. Примеры неориентированного (*а*) и ориентированного (*б*) деревьев

В случае неориентированного дерева (рис. 2.5, *а*) любая из вершин графа может быть выбрана в качестве корня. Подобный выбор определяется специфическими особенностями решаемой задачи. Так, вершина v_1 может рассматриваться в качестве корня неориентированного дерева, поскольку между v_1 и любой другой вершиной дерева всегда существует единственная простая цепь по определению (или, что менее строго, единственный неориентированный путь).

Для случая ориентированного дерева (рис. 2.5, *б*) вершина v_2 является единственным его корнем и имеет специальное обозначение v_0. Единственность

корня в ориентированном дереве следует из того факта, что ориентированный путь всегда имеет единственную вершину, которая является его началом. Поскольку в теории графов имеет значение только наличие или отсутствие связей между отдельными вершинами, деревья, как правило, изображаются специальным образом в виде иерархической структуры. При этом корень дерева изображается самой верхней вершиной в данной иерархии. Далее следуют вершины уровня 1, которые связаны с корнем одним ребром или одной дугой. Следующий уровень будет иметь номер 2, поскольку соответствующие вершины должны быть связаны с корнем двумя последовательными ребрами или дугами. Процесс построения иерархического дерева продолжается до тех пор, пока не будут рассмотрены вершины, не связанные с иными, кроме рассмотренных, или из которых не выходит ни одна дуга. В этом случае самые нижние вершины иногда называют *листьями* дерева. Важно иметь в виду, что в теории графов дерево "растет" вниз, а не вверх.

Изображенные выше деревья (рис. 2.5) можно преобразовать к виду иерархий. Например, неориентированное дерево (рис. 2.5, *а*) может быть представлено в виде иерархического дерева следующим образом (рис. 2.6, *а*). В этом случае корнем иерархии является вершина v_1. Ориентированное дерево (рис. 2.5, *б*) также может быть изображено в форме иерархического дерева (рис. 2.6, *б*), однако такое представление является единственным.

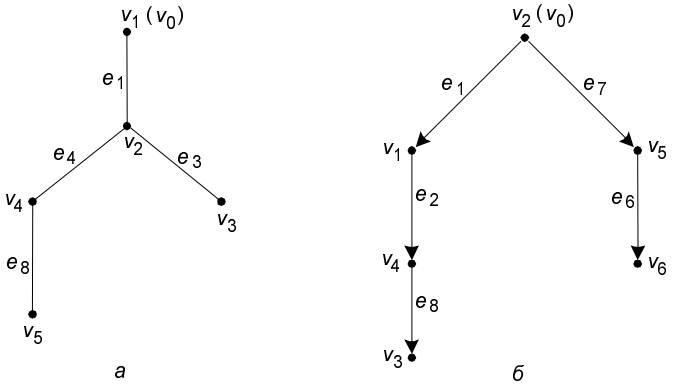


Рис. 2.6. Иерархические схемы неориентированного дерева (*а*) и ориентированного дерева (*б*)

В первом случае (рис. 2.6, *а*) вершина v_2 образует первый уровень иерархии, v_4 и v_3 — второй уровень иерархии, v_5 — третий и последний уровень иерархии. При этом листьями данного неориентированного дерева являются вершины v_3 и v_5. Во втором случае (рис. 2.6, *б*) вершины v_1 и v_5 образуют первый уровень иерархии, v_4 и v_6 — второй уровень иерархии, v_3 — третий и последний уровень иерархии. Листьями данного ориентированного дерева являются вершины v_3 и v_6.

В заключение следует заметить, что в теории графов разработаны различные методы анализа отдельных классов графов и алгоритмы построения специальных графических объектов, рассмотрение которых выходит за рамки данной книги. Для получения дополнительной информации по данной теме можно рекомендовать обратиться к специальной литературе по теории графов, где данные вопросы рассмотрены более подробно. В дальнейшем нас будет интересовать отдельное направление в теории графов, которое связано с явным включением семантики в традиционные обозначения и получившее самостоятельное развитие в форме семантических сетей.

2.1.3. Семантические сети

Семантические сети получили свое развитие в рамках научного направления, связанного с представлением знаний для моделирования рассуждений человека. Эта область научных исследований возникла в рамках общей проблематики искусственного интеллекта и была ориентирована на разработку специальных языков и графических средств для представления декларативных или, что менее точно, статических знаний о предметной области. Результаты исследований, в области семантических сетей, в последующем были конкретизированы и успешно использованы при построении концептуальных моделей и схем реляционных баз данных.

В общем случае под *семантической сетью* понимают некоторый граф $G_s = (V_s, E_s)$, в котором множества вершин V_s и ребер E_s разделены на отдельные типы, обладающие специальной семантикой, характерной для той или иной предметной области. В данной ситуации множество вершин может соответствовать объектам или сущностям рассматриваемой предметной области, и иметь вместо номеров вершин соответствующие явные имена этих сущностей. Подобные имена должны позволять однозначно идентифицировать соответствующие объекты, при этом общих формальных правил записи имен не существует. Множество ребер также делится на различные типы, которые соответствуют различным видам связей между сущностями рассматриваемой предметной области.

Так, при построении семантической сети для представления знаний о рабочем персонале некоторой компании, в качестве объектов целесообразно выбрать отдельных сотрудников, каждого из которых идентифицировать собственным именем и фамилией. Дополнительно, в сети могут присутствовать такие объекты, как рабочие проекты и подразделения компании. В качестве семантических связей можно выделить такие виды, как должностное подчинение сотрудников, их участие в работе над проектами, принадлежность сотрудников тому или иному подразделению компании.

Важной особенностью семантических сетей является разработка специальных графических обозначений для представления отдельных типов вершин и ребер. При этом вершины не изображаются, как ранее — точками, а имеют

вид прямоугольников, овалов, окружностей и других геометрических фигур, конкретный вид которых определяет тот или иной тип сущностей предметной области. Более разнообразным становится и изображение ребер, приобретающих вид различных линий со стрелками или без них, а также имеющих специальные обозначения или украшения в виде условных значков. Соответствующая система обозначений, предназначенная для представления информации об отдельных аспектах моделируемой предметной области, получила название *графической нотации*.

Примечание

В этой связи следует заметить, что различные виды диаграмм языка UML в общем случае являются специальными классами семантических сетей с достаточно развитой семантикой используемых условных обозначений. При этом унифицированный характер этих обозначений определяет конструктивность их использования для моделирования широкого круга приложений.

В качестве конкретного варианта представления информации в виде семантической сети рассмотрим дальнейшее развитие примера с классом "Автомобиль" из *главы 1*. Фрагмент семантической сети, которая описывает иерархию классов данной предметной области, может быть изображен следующим образом (рис. 2.7). На данном рисунке отдельные вершины семантической сети изображаются прямоугольниками с закругленными концами и служат для условного обозначения классов данной предметной области. Соединяющие вершины ребра имеют вполне определенный смысл или семантику. А именно, они явно указывают, что вершина или класс, расположенные на рисунке ниже, являются подклассом того класса уровнем выше, с которым имеется связь в форме соединяющего их ребра.

Например, классы "Легковой автомобиль" и "Грузовой автомобиль" являются подклассами класса "Автомобиль", а "Модель ВАЗ-21083" и "Модель ВАЗ-21099" являются подклассами класса "Легковой автомобиль производства ВАЗ". Ребра или связи данной семантической сети имеют единственный тип, определяемый семантикой включения классов друг в друга. Поэтому никаких дополнительных обозначений они не содержат.

Примечание

Изображенный выше фрагмент семантической сети может быть расширен различным образом, что определяется спецификой решаемой задачи. С одной стороны, можно ввести в рассмотрение дополнительные модели автомобилей, а с другой — другие типы объектов, например, конкретные заводы, расположенные в различных регионах, или станции, осуществляющие техническое обслуживание автомобилей. В последнем случае появляются дополнительные связи, которые могут соответствовать совершенно иной семантике. Например, факт обслуживания той или иной модели автомобиля на отдельных станциях.

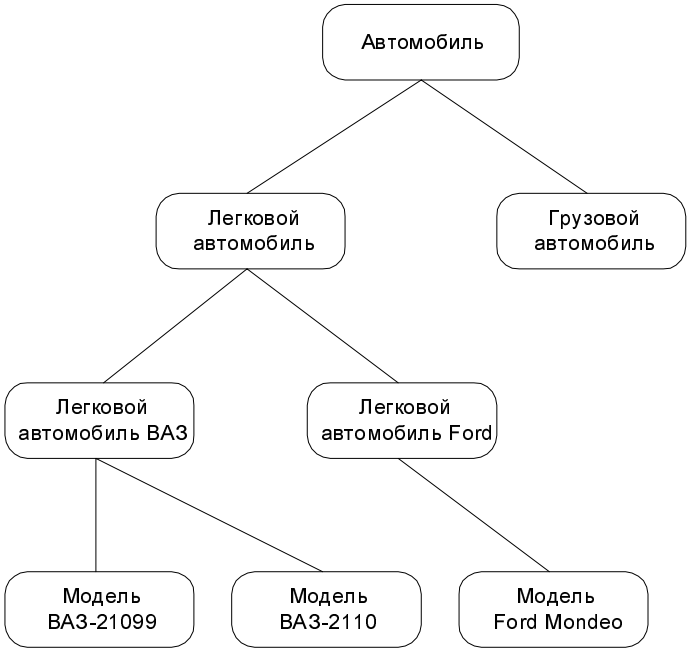


Рис. 2.7. Фрагмент семантической сети для представления иерархии классов "Автомобиль"

Построение моделей сложных систем, отражающих десятки различных типов объектов и связей между ними, привело, в конце 80-х годов прошлого столетия, к появлению большого числа различных графических нотаций, которые в той или иной степени были ориентированы на решение специальных классов задач. Сложилась парадоксальная ситуация, которая получила название "войны методов". Многие подходы, хотя и имели общие истоки, совершенно игнорировали другие альтернативные способы представления семантической информации. Наибольшее распространение в эти годы получил подход к моделированию программных систем, который получил название системный структурный анализ (ССА). Поскольку многие идеи ССА оказали непосредственное влияние на развитие языка UML, а используемая графическая нотация была реализована в некоторых CASE-средствах, далее приводится краткая характеристика основных компонентов данного направления графического моделирования программных систем.

2.2. Диаграммы структурного системного анализа

Под структурным системным анализом принято понимать метод исследования системы, который начинается с наиболее общего ее описания, с последующей детализацией представления отдельных аспектов ее поведения