

Отношения и их графическое изображение на диаграмме классов

Ключевые слова: [UML](#), [ассоциация](#), [association relationship](#), [обобщение](#), [generalization relationship](#), [агрегация](#), [композиция](#), [представление](#), [отношение](#), [диаграммы вариантов использования](#), [семантика](#), [кратность](#), [binary](#), [association](#), [класс](#), [объект](#), [кортеж](#), [XOR](#), [арность](#), [значение](#), [операции](#), [association class](#), [end](#), [имя роли](#), [интервал](#), [место](#), [наследование](#), [класс-потомок](#), [класс-предок](#), [суперкласс](#), [подкласс](#), [диаграммы классов](#), [корнем дерева](#), [произвольное](#), [Дополнение](#), [incomplete](#), [non-overlapping](#), [диаграмма](#), [очередь](#), [разбиение](#), [деление](#), [процессор](#), [плата](#), [память](#), [диск](#), [меню](#), [производительность](#), [опыт](#), [предметной области](#)

Кроме внутреннего устройства классов важную роль при разработке проектируемой системы имеют различные отношения между классами, которые также могут быть изображены на диаграмме классов. Совокупность допустимых типов таких отношений строго фиксирована в языке UML и определяется самой семантикой этих отношений. Базовые отношения, изображаемые на диаграммах классов:

- Отношение ассоциации (association relationship)
- Отношение обобщения (generalization relationship)
- Отношение агрегации (aggregation relationship)
- Отношение композиции (composition relationship)

Каждое из этих отношений имеет собственное графическое представление, которое отражает семантический характер взаимосвязи между объектами соответствующих классов.

Отношение ассоциации

Ассоциация (association) - семантическое отношение между двумя и более классами, которое специфицирует характер связи между соответствующими экземплярами этих классов.

Отношение ассоциации соответствует наличию произвольного отношения или взаимосвязи между классами. Данное отношение, как уже отмечалось в лекции 3, обозначается сплошной линией со стрелкой или без нее и с дополнительными символами, которые характеризуют специальные свойства ассоциации. Ассоциации рассматривались при изучении элементов диаграммы вариантов использования, применительно к диаграммам классов, тем не менее, семантика этого типа отношений значительно шире. В качестве дополнительных специальных символов могут использоваться имя ассоциации, символ навигации, а также имена и кратность классов-ролей ассоциации.

Имя ассоциации - необязательный элемент ее обозначения. Однако если оно задано, то записывается с заглавной буквы рядом с линией ассоциации. Отдельные классы ассоциации могут играть определенную роль в соответствующем отношении, на что явно указывает имя конечных точек ассоциации на диаграмме.

Наиболее простой случай данного отношения - **бинарная ассоциация** (binary association), которая служит для представления произвольного отношения между двумя классами. Она связывает в точности два различных класса и может быть ненаправленным (симметричным) или направленным отношением. Частный случай бинарной ассоциации - **рефлексивная ассоциация**, которая связывает класс с самим собой. Ненаправленная бинарная ассоциация изображается линией без

стрелки. Для нее на диаграмме может быть указан порядок чтения классов с использованием значка в форме треугольника рядом с именем данной ассоциации.

В качестве простого примера ненаправленной бинарной ассоциации можно рассмотреть отношение между двумя классами - классом Компания и классом Сотрудник (рис. 3-2.1). Они связаны между собой бинарной ассоциацией Работает, имя которой указано на рисунке рядом с линией ассоциации. Для данного отношения определен следующий порядок чтения следования классов - сотрудник работает в компании.

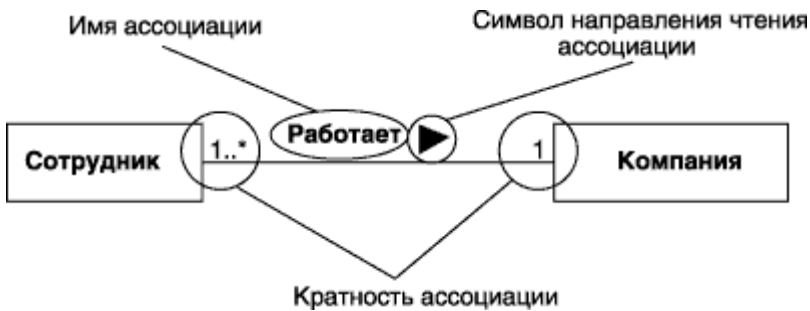


Рис. 3-2.1. Графическое изображение ненаправленной бинарной ассоциации между классами

Направленная бинарная ассоциация изображается сплошной линией с простой стрелкой на одной из ее концевых точек. Направление этой стрелки указывает на то, какой класс является первым, а какой - вторым.

В качестве простого примера направленной бинарной ассоциации можно рассмотреть отношение между двумя классами - классом Клиент и классом Счет (рис. 3-2.2). Они связаны между собой бинарной ассоциацией с именем Имеет, для которой определен порядок следования классов. Это означает, что конкретный объект класса Клиент всегда должен указываться первым при рассмотрении взаимосвязи с объектом класса Счет. Другими словами, эти объекты классов образуют кортеж элементов, например, <клиент, счет_1, счет_2,..., счет_n>.



Рис. 3-2.2. Графическое изображение направленной бинарной ассоциации между классами

Частный случай отношения ассоциации - так называемая исключаящая ассоциация (Xor-association). Семантика данной ассоциации указывает на то, что из нескольких потенциально возможных вариантов данной ассоциации в каждый момент времени может использоваться только один. На диаграмме классов исключаящая ассоциация изображается пунктирной линией, соединяющей две и более ассоциации (рис. 3-2.3), рядом с которой записывается ограничение в форме строки текста в фигурных скобках: {xor}.

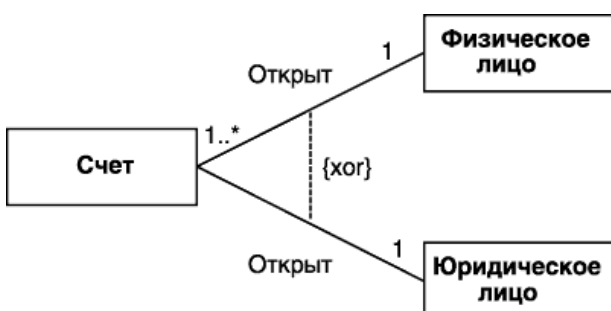


Рис. 3-2.3. Графическое изображение исключяющей ассоциации между тремя классами

Тернарная ассоциация связывает отношением три класса. Ассоциация более высокой арности называется n-арной ассоциацией.

n-арная ассоциация (n-ary association) - ассоциация между тремя и большим числом классов.

Каждый экземпляр такой ассоциации представляет собой упорядоченный набор (кортеж), содержащий n экземпляров из соответствующих классов. Такая ассоциация связывает отношением более чем три класса, при этом класс может участвовать в ассоциации более чем один раз. Каждый экземпляр n-арной ассоциации представляет собой n-арный кортеж, состоящий из объектов соответствующих классов. В этом контексте бинарная ассоциация является частным случаем n-арной ассоциации, когда значение $n=2$, но имеет собственное обозначение. Бинарная ассоциация - это специальный случай n-арной ассоциации.

Графически n-арная ассоциация обозначается ромбом, от которого ведут линии к символам классов данной ассоциации. Сам же ромб соединяется с символами классов сплошными линиями. Обычно линии проводятся от вершин ромба или от середины его сторон. Имя n-арной ассоциации записывается рядом с ромбом соответствующей ассоциации. Однако порядок классов в n-арной ассоциации, в отличие от порядка множеств в отношении, на диаграмме не фиксируется.

В качестве примера тернарной ассоциации можно рассмотреть отношение между тремя классами: Сотрудник, Компания и Проект. Данная ассоциация указывает на наличие отношения между этими тремя классами, которое может представлять информацию о проектах, реализуемых в компании, и о сотрудниках, участвующих в выполнении отдельных проектов ([рис. 3-2.4](#)).



Рис. 3-2.4. Графическое изображение тернарной ассоциации между тремя классами

Класс может быть присоединен к линии ассоциации пунктирной линией. Это означает, что данный класс обеспечивает поддержку свойств соответствующей n-арной ассоциации, а сама n-арная ассоциация имеет атрибуты, операции и/или ассоциации. Другими словами, такая ассоциация является классом с соответствующим обозначением в виде прямоугольника и самостоятельным элементом языка UML - ассоциативным классом (Association Class).

Класс ассоциация (association class) - модельный элемент, который одновременно является ассоциацией и классом. Ассоциация класс может рассматриваться как ассоциация, которая обладает свойствами класса, или как класс, имеющий также свойства ассоциации.

Как уже упоминалось, отдельный класс в ассоциации может играть определенную роль в данной ассоциации. Эта роль может быть явно специфицирована на диаграмме классов. С этой целью в языке UML вводится в рассмотрение специальный элемент - концевая точка ассоциации или конец ассоциации (Association End), который графически соответствует точке соединения линии ассоциации с отдельным классом.

Конец ассоциации (association end) - концевая точка ассоциации, которая связывает ассоциацию с классификатором.

Конец ассоциации - часть ассоциации, но не класса. Каждая ассоциация может иметь два или больше концов ассоциации. Наиболее важные свойства ассоциации указываются на диаграмме рядом с этими элементами ассоциации и должны перемещаться вместе с ними. Одним из таких дополнительных обозначений является имя роли отдельного класса, входящего в ассоциацию.

Роль (role) - имеющее имя специфическое поведение некоторой сущности, рассматриваемой в определенном контексте. Роль может быть статической или динамической.

Имя роли представляет собой строку текста рядом с концом ассоциации для соответствующего класса. Она указывает на специфическую роль, которую играет класс, являющийся концом рассматриваемой ассоциации. Имя роли не обязательный элемент обозначений и может отсутствовать на диаграмме.

Следующий элемент обозначений - кратность ассоциации. Кратность относится к концам ассоциации и обозначается в виде интервала целых чисел, аналогично кратности атрибутов и операций классов, но без прямых скобок. Этот интервал записывается рядом с концом соответствующей ассоциации и означает потенциальное число отдельных экземпляров класса, которые могут иметь место, когда остальные экземпляры или объекты классов фиксированы.

Так, для примера (рис. 3-2.4) кратность " 1 " для класса Компания означает, что каждый сотрудник может работать только в одной компании. Кратность " 1..* " для класса Сотрудник означает, что в каждой компании могут работать несколько сотрудников, общее число которых заранее неизвестно и ничем не ограничено. Вместо кратности " 1..* " нельзя записать только символ " * ", поскольку последний означает кратность " 0..* ". Для данного примера это означало бы, что отдельные компании могут совсем не иметь сотрудников в своем штате. Такая кратность приемлема в ситуациях, когда в компании вообще не выполняется никаких проектов.

Имя ассоциации рассматривается в качестве отдельного атрибута у соответствующих классов ассоциаций и может быть указано на диаграмме явным способом в определенной секции прямоугольника класса.

Ассоциация является наиболее общей формой отношения в языке UML. Все другие типы рассматриваемых отношений можно считать частным случаем данного отношения. Однако важность выделения специфических семантических свойств и дополнительных характеристик для других типов отношений обуславливают необходимость их самостоятельного изучения при построении диаграмм классов. Поскольку эти отношения имеют специальные обозначения и относятся к базовым понятиям языка UML, следует перейти к их последовательному рассмотрению.

Отношение обобщения

Отношение обобщения является обычным таксономическим отношением или отношением классификации между более общим элементом (родителем или предком) и более частным или специальным элементом (дочерним или потомком).

Обобщение (generalization) - таксономическое отношение между более общим понятием и менее общим понятием.

Менее общий элемент модели должен быть согласован с более общим элементом и может содержать дополнительную информацию. Данное отношение используется для представления

иерархических взаимосвязей между различными элементами языка UML, такими как пакеты, классы, варианты использования.

Применительно к диаграмме классов данное отношение описывает иерархическое строение классов и наследование их свойств и поведения.

Наследование (inheritance) - специальный концептуальный механизм, посредством которого более специальные элементы включают в себя структуру и поведение более общих элементов.

Согласно одному из главных принципов методологии ООАП - наследованию, класс-потомок обладает всеми свойствами и поведением класса-предка, а также имеет собственные свойства и поведение, которые могут отсутствовать у класса-предка.

Родитель, предок (parent) - в отношении обобщения более общий элемент. Потомок (child) - специализация одного из элементов отношения обобщения, называемого в этом случае родителем.

На диаграммах отношение обобщения обозначается сплошной линией с треугольной стрелкой на одном из концов ([рис. 3-2.5](#)). Стрелка указывает на более общий класс (класс-предок или суперкласс), а ее начало - на более специальный класс (класс-потомок или подкласс).



Рис. 3-2.5. Графическое изображение отношения обобщения в языке UML

От одного класса-предка одновременно могут наследовать несколько классов-потомков, что отражает таксономический характер данного отношения. В этом случае на диаграмме классов для подобного отношения обобщения указывается несколько линий со стрелками.

Например, класс *Транспортное средство* (курсив обозначает абстрактный класс) может выступать в качестве суперкласса для подклассов, соответствующих конкретным транспортным средствам, таким как: Автомобиль, Автобус, Трактор и другим. Это может быть представлено графически в форме диаграммы классов следующего вида ([рис. 3-2.6](#)).



Рис. 3-2.3-2. Пример графического изображения отношения обобщения для нескольких классов-потомков

С целью упрощения обозначений на диаграмме классов и уменьшения числа стрелок-треугольников и совокупности линий, обозначающих одно и то же отношение обобщения, может быть просто

изображена стрелка. В этом случае отдельные линии сходятся к стрелке, которая имеет с этими линиями единственную точку пересечения ([рис. 3-2.7](#)).

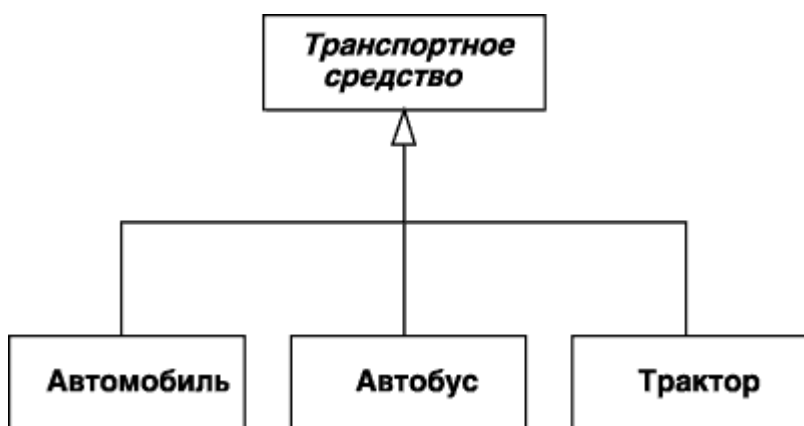


Рис. 3-2.7. Альтернативный вариант графического изображения отношения обобщения классов для случая объединения отдельных линий

Графическое изображение отношения обобщения по форме соответствует графу специального вида, а именно - иерархическому дереву. Как нетрудно заметить, в этом случае класс-предок является корнем дерева, а классы-потомки - его листьями. Отличие заключается в возможности указания на диаграмме классов дополнительной семантической информации, которая может отражать различные теоретико-множественные характеристики данного отношения. При этом класс-предок на диаграмме может занимать произвольное положение относительно своих классов-потомков, определяемое лишь соображениями удобства.

В дополнение к простой стрелке обобщения может быть присоединена строка текста, указывающая на специальные свойства этого отношения в форме ограничения. Этот текст будет относиться ко всем линиям обобщения, которые идут к классам-потомкам. Поскольку отмеченное свойство касается всех подклассов данного отношения, именно поэтому спецификация этого свойства осуществляется в форме ограничения, которое должно быть записано в фигурных скобках.

В качестве ограничений могут быть использованы следующие ключевые слова языка UML:

- `{complete}` - означает, что в данном отношении обобщения специфицированы все классы-потомки, и других классов-потомков у данного класса-предка быть не может.
- `{incomplete}` - означает случай, противоположный первому. А именно, предполагается, что на диаграмме указаны не все классы-потомки. В последующем возможно разработчик восполнит их перечень, не изменяя уже построенную диаграмму.
- `{disjoint}` - означает, что классы-потомки не могут содержать объектов, одновременно являющихся экземплярами двух или более классов.
- `{overlapping}` - случай, противоположный предыдущему. А именно, предполагается, что отдельные экземпляры классов-потомков могут принадлежать одновременно нескольким классам.

С учетом дополнительного использования стандартного ограничения диаграмма классов ([рис. 3-2.7](#)) может быть уточнена ([рис. 3-2.8](#)).

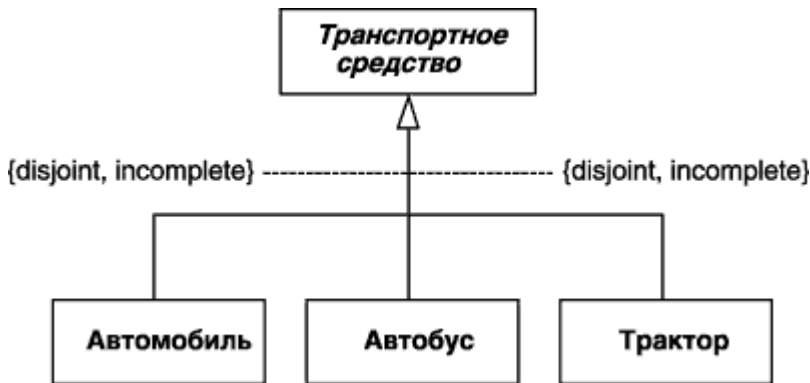


Рис. 3-2.8. Вариант уточненного графического изображения отношения обобщения классов с использованием строки-ограничения

Отношение агрегации

Агрегация (aggregation) - специальная форма ассоциации, которая служит для представления отношения типа "часть-целое" между агрегатом (целое) и его составной частью.

Отношение агрегации имеет место между несколькими классами в том случае, если один из классов представляет собой сущность, которая включает в себя в качестве составных частей другие сущности. Данное отношение имеет фундаментальное значение для описания структуры сложных систем, поскольку применяется для представления системных взаимосвязей типа "часть-целое". Раскрывая внутреннюю структуру системы, отношение агрегации показывает, из каких элементов состоит система, и как они связаны между собой.

С точки зрения модели отдельные части системы могут выступать в виде, как элементов, так и подсистем, которые, в свою очередь, тоже могут состоять из подсистем или элементов. Таким образом, данное отношение по своей сути описывает декомпозицию или разбиение сложной системы на более простые составные части, которые также могут быть подвергнуты декомпозиции, если в этом возникнет необходимость.

Очевидно, что рассматриваемое в таком аспекте деление системы на составные части представляет собой иерархию, но принципиально отличную от той, которая порождается отношением обобщения. Отличие заключается в том, что части системы никак не обязаны наследовать ее свойства и поведение, поскольку являются самостоятельными сущностями. Более того, части целого обладают собственными атрибутами и операциями, которые существенно отличаются от атрибутов и операций целого.

Графически отношение агрегации изображается сплошной линией, один из концов которой представляет собой не закрашенный внутри ромб. Этот ромб указывает на тот класс, который представляет собой "целое" или класс-контейнер. Остальные классы являются его "частями" (рис. 3-2.9).



Рис. 3-2.9. Графическое изображение отношения агрегации в языке UML

В качестве примера отношения агрегации можно рассмотреть взаимосвязь типа "часть-целое", которая имеет место между классом Системный блок персонального компьютера и его составными частями: Процессор, Материнская плата, Оперативная память, Жесткий диск и Дисковод гибких

дисков. Используя обозначения языка UML, компонентный состав системного блока можно представить в виде соответствующей диаграммы классов (рис. 3-2.10), которая в данном случае иллюстрирует отношение агрегации.



Рис. 3-2.10. Диаграмма классов для иллюстрации отношения агрегации на примере системного блока ПК

Отношение композиции

Композиция (composition) - разновидность отношения агрегации, при которой составные части целого имеют такое же время жизни, что и само целое. Эти части уничтожаются вместе с уничтожением целого.

Отношение композиции - частный случай отношения агрегации. Это отношение служит для спецификации более сильной формы отношения "часть-целое", при которой составляющие части тесно взаимосвязаны с целым. Особенность этой взаимосвязи заключается в том, что части не могут выступать в отрыве от целого, т.е. с уничтожением целого уничтожаются и все его составные части.

Композит (composite) - класс, который связан отношением композиции с одним или большим числом классов.

Графически отношение композиции изображается сплошной линией, один из концов которой представляет собой закрашенный внутри ромб. Этот ромб указывает на тот класс, который представляет собой класс-композит. Остальные классы являются его "частями" (рис. 3-2.11).



Рис. 3-2.11. Графическое изображение отношения композиции в языке UML

Возможно, самым наглядным примером этого отношения является живая клетка в биологии, в отрыве от которой не могут существовать ее составные части. Другой практический пример - окно графического интерфейса программы, которое может состоять из строки заголовка, полос прокрутки, главного меню, рабочей области и строки состояния. Нетрудно заключить, что подобное окно представляет собой класс, а его составные элементы также являются отдельными классами. Последнее обстоятельство характерно для отношения композиции, поскольку отражает различные способы представления данного отношения.

Для отношений композиции и агрегации могут использоваться дополнительные обозначения, применяемые для отношения ассоциации. А именно, могут указываться кратности отдельных классов, которые в общем случае не обязательны. Применительно к описанному выше примеру класс Окно программы является классом-композитом, а взаимосвязи составляющих его частей могут быть изображены следующей диаграммой классов (рис. 3-2.12).

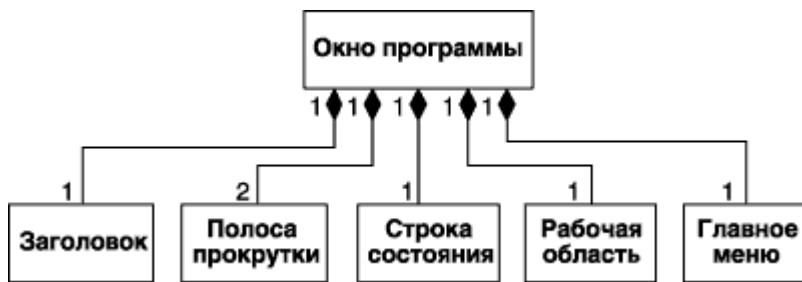


Рис. 3-2.12. Диаграмма классов для иллюстрации отношения композиции на примере класса-композиита Окно программы

Рекомендации по построению диаграмм классов

Процесс разработки диаграммы классов занимает центральное место при разработке проектов сложных систем. От умения правильно выбрать классы и установить между ними взаимосвязи часто зависит не только успех процесса проектирования, но и производительность выполнения программы. Как показывает практика ООАП, каждый программист в той или иной степени использует личный опыт при разработке новых проектов. Это обусловлено желанием свести новую задачу к уже решенным, чтобы иметь возможность применять не только проверенные фрагменты программного кода, но и отдельные компоненты или библиотеки компонентов.

При определении классов, атрибутов и операций, задании их имен и типов перед российскими разработчиками всегда встает вопрос: какой язык взять за основу, русский или английский? С одной стороны, использование родного языка для описания модели кажется наиболее естественным способом ее представления и в наибольшей степени отражает коммуникативную функцию модели системы. С другой стороны, разработка модели всего лишь один из этапов создания соответствующей системы, а применение инструментальных средств для ее реализации в абсолютном большинстве случаев требует использования англоязычных терминов. Именно поэтому возникает характерная неоднозначность, с которой, по-видимому, совершенно незнакома англоязычная аудитория.

При построении диаграммы вариантов использования, общей концептуальной модели проектируемой системы, применение русскоязычных терминов оправдано с точки зрения описания структуры предметной области. Это способствует взаимопониманию, а также эффективному общению между заказчиком и разработчиком проекта. При построении остальных типов диаграмм следует придерживаться разумного компромисса.

После разработки диаграммы классов процесс ООАП может быть продолжен в двух направлениях. С одной стороны, если поведение системы тривиально, то можно приступить к разработке диаграмм кооперации и последовательности. Особенности построения диаграмм кооперации будут рассмотрены в следующей главе. Однако для сложных динамических систем, в частности для параллельных систем и систем реального времени, важнейшим аспектом их функционирования является поведение. Специфика поведения таких систем может быть представлена на диаграммах состояний и деятельности, разработка которых в общем случае не обязательна, а определяется конкретным проектом.