

ДИАГРАММА КЛАССОВ

(CLASS DIAGRAM)



Диаграмма классов — тип статической структуры модели системы, которая может отражать различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами.

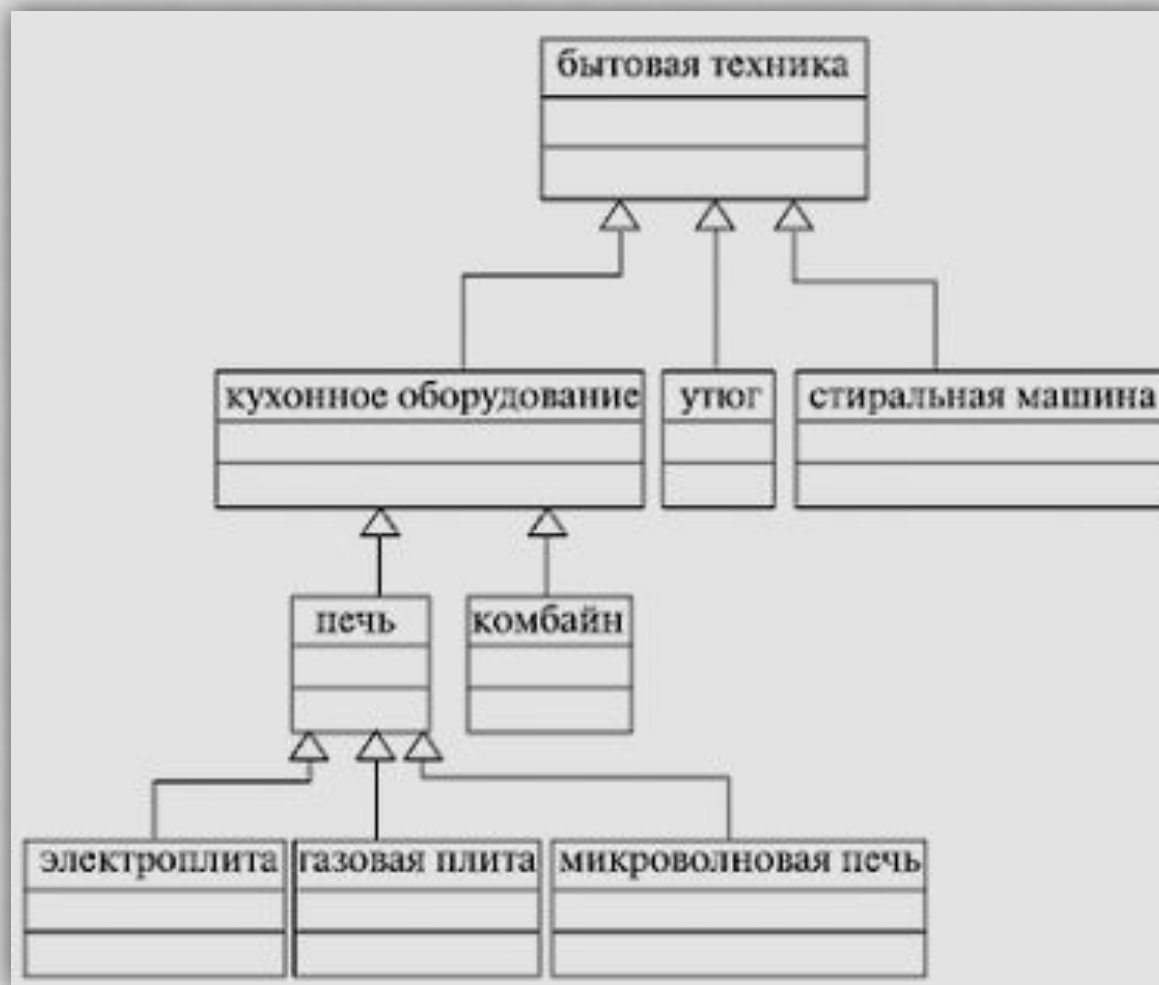


Рис.1. Пример диаграммы классов, которая иллюстрирует с помощью операции наследования «генеалогическое древо» бытовой техники.

КЛАСС (CLASS)

Класс в языке UML служит для обозначения множества объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других классов. Класс на диаграмме изображается в виде прямоугольника, разделенного горизонтальными линиями на три части, где указываются имя класса, атрибуты и операции (методы).



Рис.2. Пример структуры класса

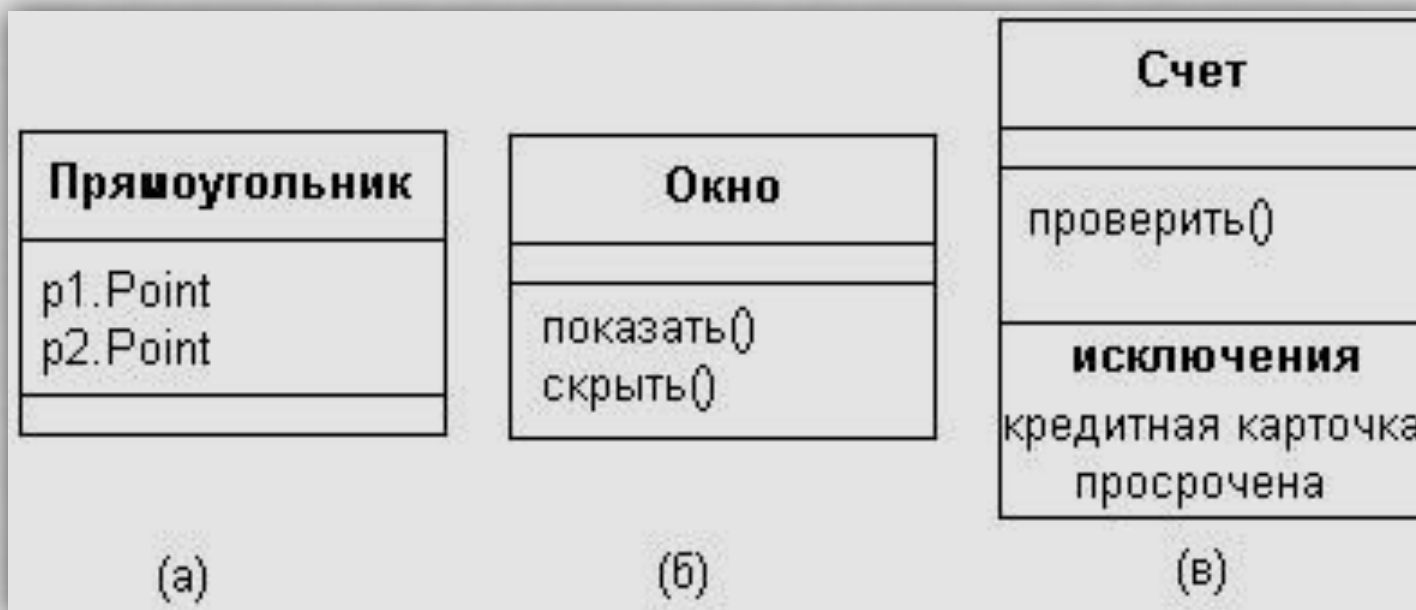


Рис.3. Пример классов

ИМЯ КЛАССА

Имя класса должно быть уникальным в пределах пакета, который описывается совокупностью диаграмм классов (возможно, одной). Оно указывается в верхней секции прямоугольника. Имя класса записывается по центру секции имени полужирным шрифтом и должно начинаться с заглавной буквы. Рекомендуется в качестве имен классов использовать существительные, записанные по практическим соображениям без пробелов. Необходимо помнить, что именно имена классов образуют словарь предметной области при ООАП.

АТТРИБУТЫ КЛАССА (ATTRIBUTES)

Атрибут описывает свойство в виде строки текста внутри прямоугольника класса.

Полная форма атрибута:

видимость имя: тип кратность = значение
по умолчанию {строка свойств}

Например:

- имя: *String* [1] = "Без имени" {readOnly}

Квантор видимости может принимать одно из трех возможных значений и, соответственно, отображается при помощи специальных символов:

- Символ "+" обозначает атрибут с областью видимости типа общедоступный (public). Атрибут с этой областью видимости доступен или виден из любого другого класса пакета, в котором определена диаграмма.
- Символ "#" обозначает атрибут с областью видимости типа защищенный (protected). Этот атрибут недоступен или невиден для всех классов, за исключением подклассов данного класса.
- И, наконец, знак "-" обозначает атрибут с областью видимости типа закрытый (private), который недоступен или невиден для всех классов без исключения.

Имя атрибута – способ ссылки класса на атрибут.

Тип атрибута накладывает ограничение на вид объекта, который может быть размещен в атрибуте. Можно считать его аналогом типа поля в языке программирования.

Значение по умолчанию представляет собой значение для вновь создаваемых объектов, если атрибут не определен в процессе создания.

Элемент {строка свойств} позволяет указывать дополнительные свойства атрибута. В примере он равен {readOnly}, то есть клиенты не могут изменять атрибут. Если он пропущен, то, как правило, атрибут можно модифицировать.

Кратность свойства обозначает количество объектов, которые могут заполнять данное свойство. Чаще всего встречаются следующие кратности:

- [1] означает, что кратность атрибута принимает значение 1.
- [0..1] означает, что кратность атрибута может принимать значение 0 или 1.
- [0..*] или просто [*] означает, что кратность атрибута может принимать любое целое значение, большее или равное 0.

В большинстве случаев кратности определяются своими нижней и верхней границами, например $[1..5]$. Нижняя граница может быть нулем или положительным числом, верхняя граница представляет собой положительное число или $*$ (без ограничений). Если нижняя и верхняя границы совпадают, то можно указать одно число; поэтому $[1]$ эквивалентно $[1..1]$. Поскольку это общий случай, $[*]$ является сокращением $[0..*]$.

ОПЕРАЦИЯ (OPERATION)

Операция представляет собой действие, реализуемое некоторым классом.

Совокупность операций характеризует функциональный аспект поведения класса.

Полный синтаксис операций в языке UML выглядит следующим образом:

*видимость имя (список параметров) :
возвращаемый тип {строка свойств}.*

Имя – это строка текста, которая используется в качестве идентификатора соответствующей операции.

Список параметров – перечень разделенных запятой формальных параметров.

Возвращаемый тип – тип возвращаемого значения, если таковое есть.

Строка свойств – значения свойств, которые применяются к данной операции.

Операция, которая не может изменять состояние системы и, соответственно, не имеет никакого побочного эффекта, обозначается строкой-свойством "{запрос}" ("{query}"). В противном случае операция может изменять состояние системы, хотя нет никаких гарантий, что она будет это делать.

ОТНОШЕНИЯ МЕЖДУ КЛАССАМИ

Базовыми отношениями в языке UML являются:

- Отношение зависимости (dependency relationship)
- Отношение ассоциации (association relationship)
- Отношение обобщения (generalization relationship)
- Отношение реализации (realization relationship)

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

ОТНОШЕНИЕ ЗАВИСИМОСТИ

Зависимость – это отношение использования, согласно которому изменение специфики одного элемента может повлиять на другой (обратная не обязательно). Обозначается - - - -> (направление стрелки на тот элемент, от которого зависит данный)

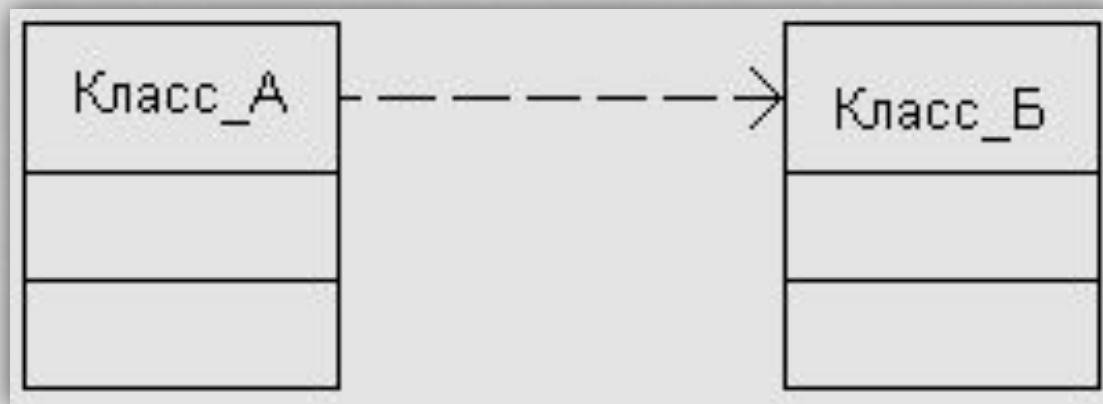


Рис.4. Графическое изображение отношения зависимости

ОТНОШЕНИЕ АССОЦИАЦИИ

Ассоциация – это структурное отношение, которое показывает, что объекты одного типа некоторым способом связаны с объектами другого типа. Обозначается сплошной линией с дополнительными специальными символами, которые характеризуют отдельные свойства конкретной ассоциации. В качестве дополнительных специальных символов могут использоваться имя ассоциации, а также имена и кратность классов-ролей ассоциации.

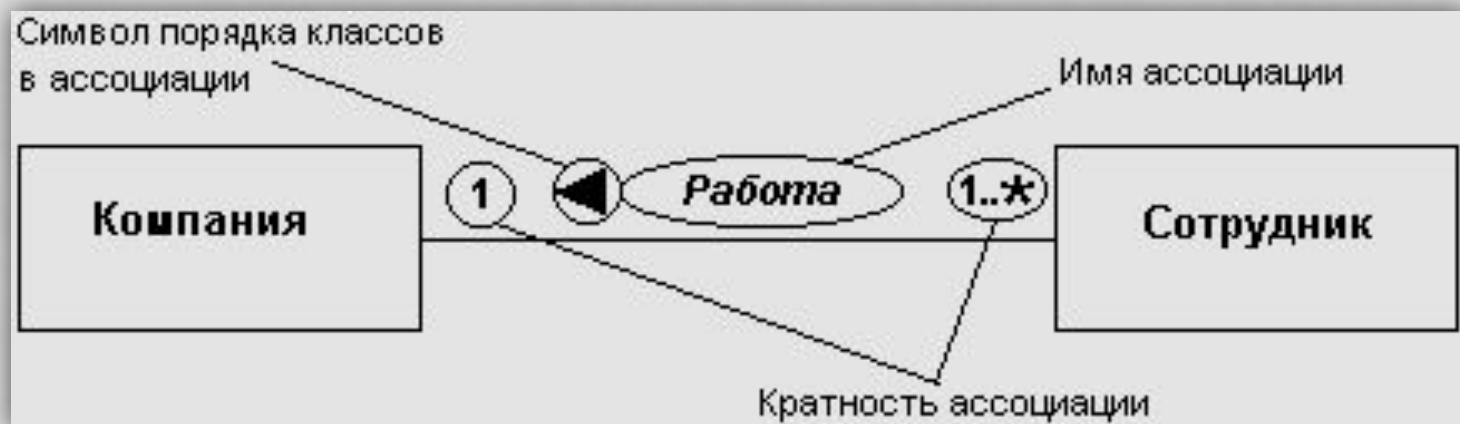


Рис.5. Графическое изображение отношения бинарной ассоциации между классами

Частным случаем отношения ассоциации является так называемая исключающая ассоциация (Xor-association). Семантика данной ассоциации указывает на тот факт, что из нескольких потенциально возможных вариантов в каждый момент времени может использоваться только один ее экземпляр. Изображается пунктирной линией, соединяющей две и более ассоциации, рядом с которой записывается строка-ограничение "{хог}".

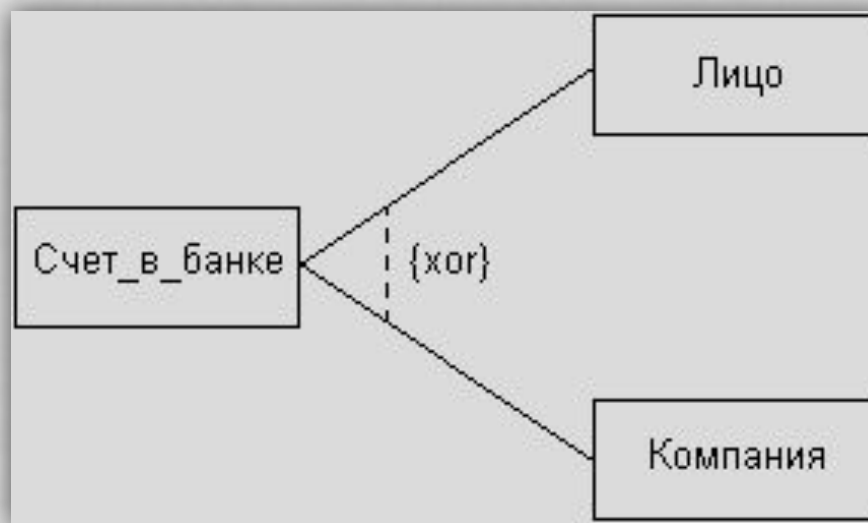


Рис.6. Графическое изображение исключяющей ассоциации между тремя классами

ОТНОШЕНИЕ АГРЕГАЦИИ

Отношение агрегации имеет место между несколькими классами в том случае, если один из классов представляет собой некоторую сущность, включающую в себя в качестве составных частей другие сущности. Раскрывая внутреннюю структуру системы, отношение агрегации показывает, из каких компонентов состоит система и как они связаны между собой.



Рис.7. Графическое изображение отношения агрегации в языке UML



Рис.8. Диаграмма классов для иллюстрации отношения агрегации на примере ПК

ОТНОШЕНИЕ КОМПОЗИЦИИ

Композиция – частный случай отношения агрегации. Она служит для выделения специальной формы отношения "часть-целое", при которой составляющие части находятся внутри целого. Специфика взаимосвязи между ними заключается в том, что части не могут выступать в отрыве от целого, т. е. с уничтожением целого уничтожаются и все его составные части.



Рис.9. Графическое изображение отношения композиции в языке UML



Рис.10. Диаграмма классов для иллюстрации отношения композиции на примере класса окна программы

ОТНОШЕНИЕ ОБОБЩЕНИЯ

Обобщение – таксономическое отношение между более общим элементом (родителем или предком) и более частным или специальным элементом (дочерним или потомком).

Применительно к диаграмме классов данное отношение описывает иерархическое строение классов и наследование их свойств и поведения. При этом предполагается, что класс-потомок обладает всеми свойствами и поведением класса-предка, а также имеет свои собственные свойства и поведение, которые отсутствуют у класса-предка.



Рис.11. Графическое изображение отношения обобщения в языке UML



Рис.12. Вариант графического изображения отношения обобщения классов

ИНТЕРФЕЙСЫ

Интерфейсы являются элементами диаграммы вариантов использования. Однако при построении диаграммы классов отдельные интерфейсы могут уточняться, и в этом случае для их изображения используется специальный графический символ — прямоугольник класса с ключевым словом или стереотипом "interface". При этом секция атрибутов у прямоугольника отсутствует, а указывается только секция операций.



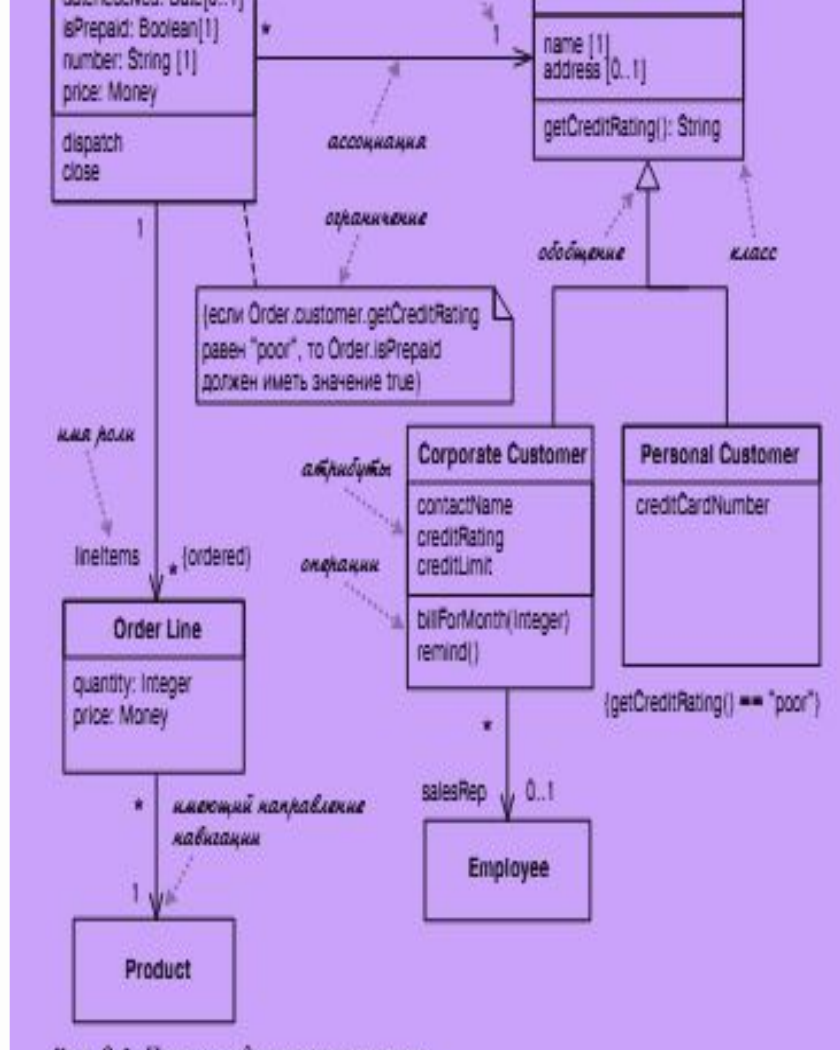
Рис.13. Пример графического изображения интерфейса на диаграмме классов

ОБЪЕКТЫ

Объект является отдельным экземпляром класса, который создается на этапе выполнения программы. Он имеет свое собственное имя и конкретные значения атрибутов. В силу самых различных причин может возникнуть необходимость показать взаимосвязи не только между классами модели, но и между отдельными объектами, реализующими эти классы. Запись имени объекта представляет собой строку текста "имя объекта:имя класса", разделенную двоеточием.



Рис.14. Пример графического изображения объектов на диаграммах языка UML



Ветчинкин В., III курс ИСВГС

ИЛ РГУ 2012