



UML 4 2016 |

Диаграмма кооперации

Поведение системы описывается на уровне отдельных объектов, которые обмениваются между собой сообщениями, чтобы достичь определенной цели или реализовать некоторый вариант использования.

Кооперация. (*collaboration*) - служит для обозначения множества взаимодействующих с определенной целью объектов в общем контексте моделируемой системы.

УРОВНИ КООПЕРАЦИИ

уровне спецификации - показывает роли классификаторов и роли ассоциаций в рассматриваемом взаимодействии;

уровне примеров - указывает экземпляры и связи, образующие отдельные роли в кооперации.

ДИАГРАММА КООПЕРАЦИИ УРОВНЯ ПРИМЕРОВ

Совокупность объектов (**экземпляры классов**) и связей (**экземпляры ассоциаций**).

Связи дополняются стрелками сообщений.

Показываются только объекты, имеющие непосредственное отношение к реализации операции или классификатора. Не обязательно изображать все свойства или все ассоциации, поскольку на диаграмме кооперации присутствуют только роли классификаторов, но не сами классификаторы.

В то время как классификатор требует полного описания всех своих экземпляров, роль классификатора требует описания только тех свойств и ассоциаций, которые необходимы для участия в отдельной кооперации.

КООПЕРАЦИЯ НА УРОВНЕ СПЕЦИФИКАЦИИ

Изображается на диаграмме пунктирным эллипсом, внутри которого записывается имя этой кооперации.

Представление кооперации относится к отдельному **варианту использования** и детализирует особенности его последующей реализации.

Символ эллипса кооперации соединяется отрезками пунктирной линии с каждым из участников этой кооперации, в качестве которых могут выступать объекты или классы.

Каждая из этих пунктирных линий помечается ролью (role) участника. Роли соответствуют именам элементов в контексте всей кооперации. Эти имена трактуются как параметры, которые ограничивают спецификацию элементов при любом их появлении в отдельных представлениях модели.

Основные компоненты

Основные компоненты диаграммы кооперации:

- объекты;**
- связи;**
- сообщения.**

КЛАССЫ

прямоугольник класса, внутри которого записывается строка текста - **роль классификатора (classifier role)**. Роль классификатора показывает особенность использования объектов данного класса.

Строка текста в прямоугольнике должна иметь следующий формат:

'/' <Имя роли классификатора> ':' <Имя классификатора>

[':' <Имя классификатора >]*

«::» имя классификатора может включать полный путь всех вложенных пакетов. один пакет от другого.

можно ограничиться указанием только ближайшего из пакетов, которому принадлежит данная кооперация.

«*» применяется для указания возможности итеративного повторения имени классификатора.

ОБЪЕКТ (ОБЪЕКТ)

Является отдельным экземпляром **класса**, который создается на этапе выполнения программы.

Может иметь свое собственное имя и конкретные значения атрибутов.

<Имя объекта> '/' <Имя роли классификатора> ':' <Имя классификатора>

[':' <Имя классификатора >]*

:C — анонимный объект, образуемый на основе класса C;

/R — анонимный объект, играющий роль R;

/R : C — анонимный объект, образуемый на основе класса C и играющий роль R;

O / R — объект с именем O, играющий роль R;

O : C — объект с именем O, образуемый на основе класса C;

O / R : C — объект с именем O, образуемый на основе класса C и играющий роль R;

O или — объект с именем O;

O : — «объект-сирота» с именем O;

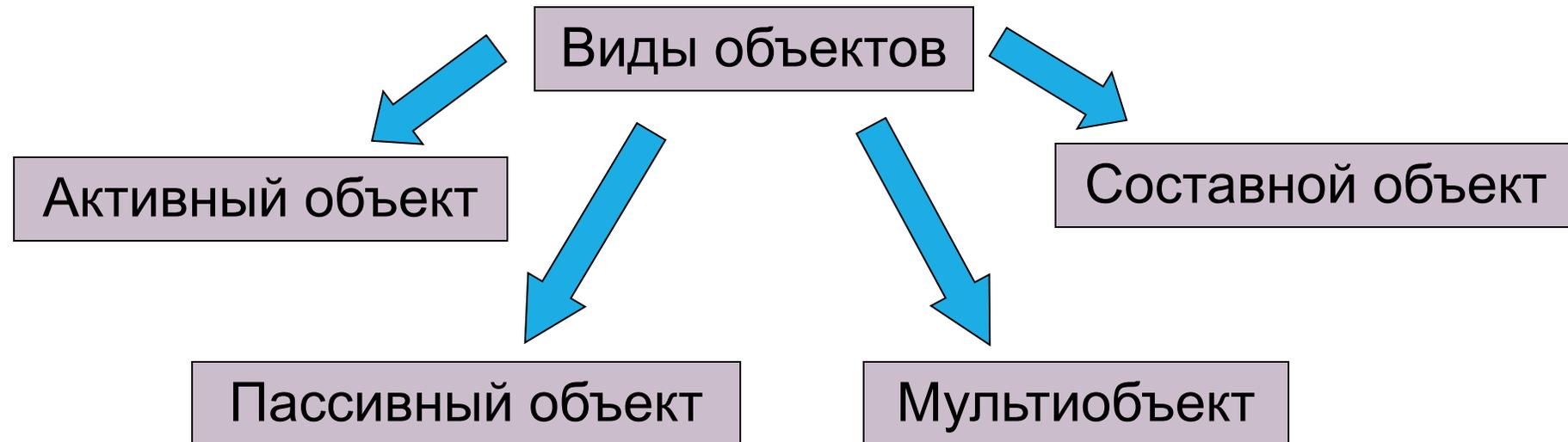
/R — роль с именем R;

/: C — анонимная роль на базе класса C;

/R : C — роль с именем R на основе класса C.

Объекты

Объект является отдельным экземпляром класса, который создается на этапе реализации модели (выполнения программы)



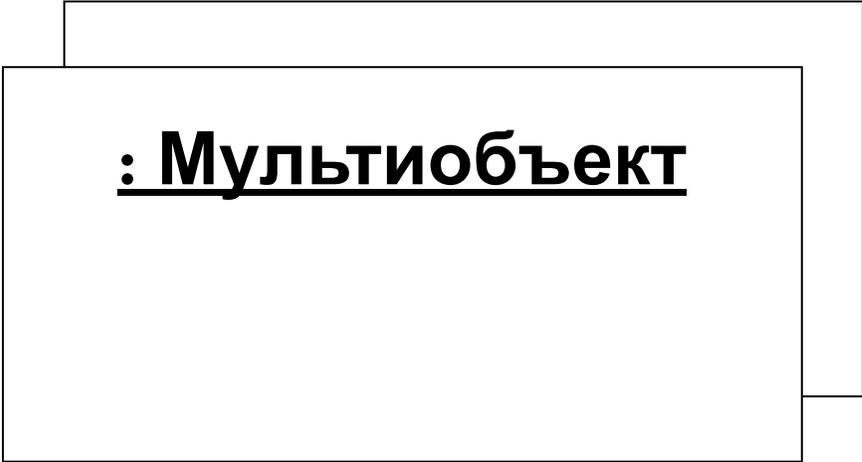
Мультиобъект

Множество объектов на одном из концов ассоциации.

Используется для того, чтобы показать операции и сигналы, которые адресованы всему множеству объектов, а не только одному.

Стрелка сообщения относится ко всему множеству объектов, которые обозначают данный мультиобъект.

Может быть явно указано отношение композиции между мультиобъектом и отдельным объектом из его множества

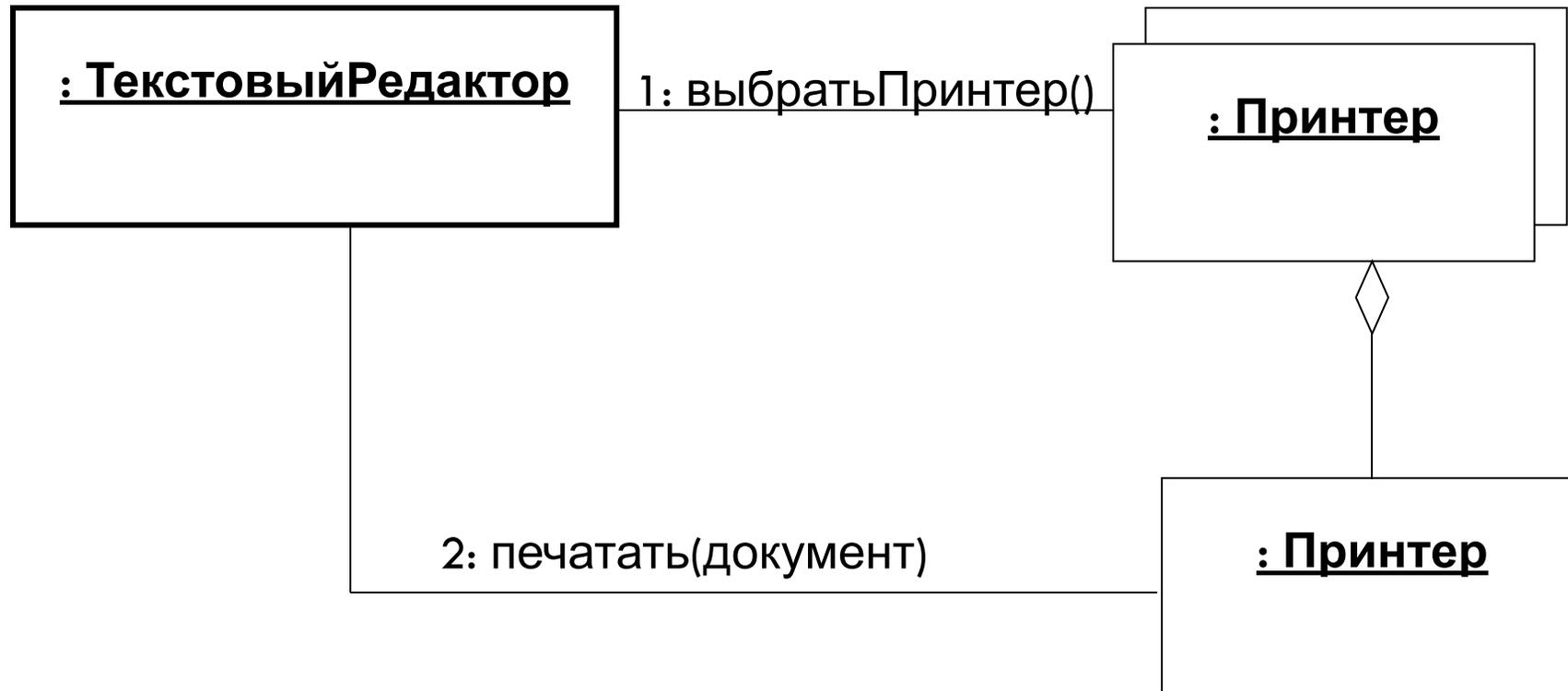


: Мультиобъект

Активный объект

В контексте языка UML объекты делятся на активные и пассивные.

Активный объект имеет свой собственный поток управления и может инициировать деятельность по управлению другими объектами.





Пассивный объект оперирует только данными и **не может инициировать деятельность по управлению другими объектами**. Однако пассивные объекты могут посылать сигналы в процессе выполнения запросов, которые они получают.

Активный объект (active object) имеет свой поток (thread) управления и **может инициировать деятельность по управлению другими объектами**.

обозначаются прямоугольником с более широкими границами.

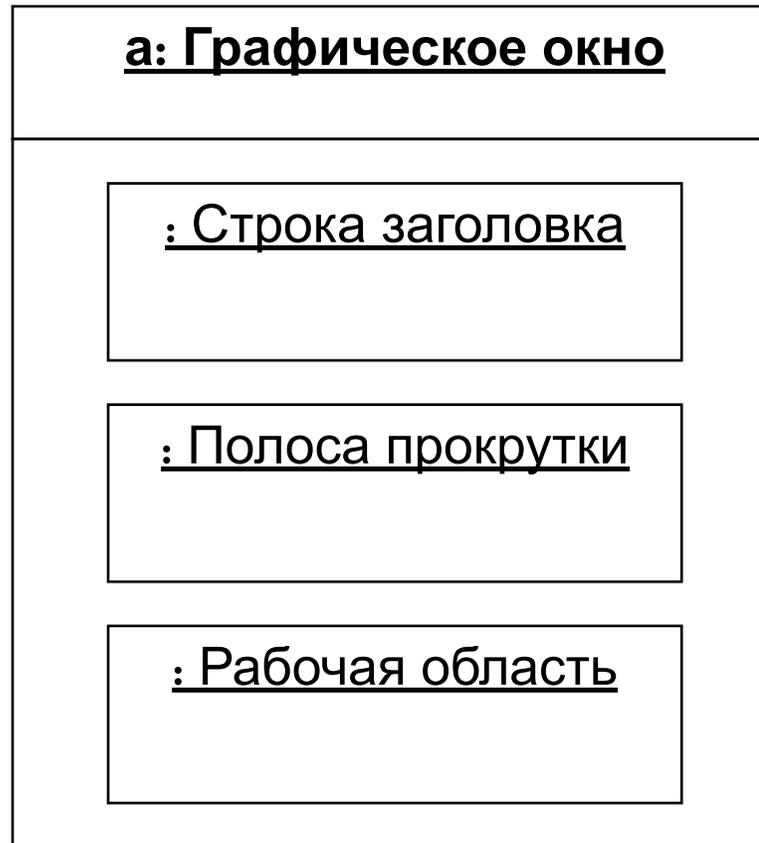
может быть явно указано ключевое слово **{active}**, чтобы выделить активный объект на диаграмме.

Каждый активный объект может инициировать единственный процесс управления и представлять исходную точку потока управления.

Составной объект (объект-контейнер)

предназначен для представления объекта, имеющего собственную структуру и внутренние потоки управления.

является экземпляром составного класса (класса-контейнера), который связан отношением агрегации или композиции со своими частями. Аналогичные отношения связывают между собой и соответствующие объекты.



СВЯЗЬ (LINK)

Экземпляр или пример произвольной ассоциации.

Бинарная связь на диаграмме кооперации изображается отрезком прямой линии, соединяющей два прямоугольника объектов.

На каждом из концов этой линии **могут быть явно указаны имена ролей данной ассоциации.**

Рядом с линией в ее средней части **может записываться имя соответствующей ассоциации.**

Связи не имеют собственных имен, поскольку полностью идентичны как экземпляры ассоциации. Другими словами, все связи на диаграмме кооперации могут быть только анонимными и записываются без двоеточия перед именем ассоциации. Для связей не указывается также и кратность. Однако другие обозначения специальных случаев ассоциации (агрегация, композиция) могут присутствовать на отдельных концах связей.

СТЕРЕОТИПЫ

«association» - ассоциация (предполагается по умолчанию, поэтому этот стереотип можно не указывать);

«parameter» - параметр метода. Соответствующий объект может быть только параметром некоторого метода;

«local» - локальная переменная метода. Ее область видимости ограничена только соседним объектом;

«global» - глобальная переменная. Ее область видимости распространяется на всю диаграмму кооперации;

«self» - рефлексивная связь объекта с самим собой, которая допускает передачу объектом сообщения самому себе. На диаграмме кооперации рефлексивная связь изображается петлей в верхней части прямоугольника объекта.

СООБЩЕНИЯ

Сообщение на диаграмме кооперации **специфицирует коммуникацию между двумя объектами**, один из которых передает другому некоторую информацию.

Первый объект ожидает, что после получения сообщения вторым объектом последует выполнение некоторого действия.

сообщение является причиной или стимулом для начала выполнения операций, отправки сигналов, создания и уничтожения отдельных объектов.

Связь обеспечивает канал для направленной передачи сообщений между объектами от объекта-источника к объекту-получателю.

Сообщения специфицируют роли, которые играют объекты отправитель и получатель сообщения.

Сообщения на диаграмме кооперации изображаются помеченными стрелками рядом (выше или ниже) с соответствующей связью или ролью ассоциации. Направление стрелки указывает на получателя сообщения. Внешний вид стрелки сообщения имеет определенный смысл.

Сообщение 2

В общем смысле под *сообщением* понимается законченный фрагмент информации, посылаемый одним объектом другому.

ФОРМАТ СООБЩЕНИЯ

< Предшествующие сообщения> < [Сторожевое условие] >

<Выражение последовательности>

<Возвращаемое значение— имя сообщения> <Список аргументов>

ПРЕДШЕСТВУЮЩИЕ СООБЩЕНИЯ

это разделенные запятыми номера сообщений, записанные перед наклонной черточкой:

<Номер сообщения ','>< Номер сообщения,'> '/'

Если список номеров сообщений пуст, то вся запись, включая наклонную черту, опускается. Каждый номер сообщения может быть выражением последовательности без рекурсивных символов. Выражение должно определять номер другого сообщения в этой же последовательности.

Смысл указания предшествующих сообщений заключается в том, что данное сообщение не может быть передано, пока не будут переданы своим адресатам все сообщения, номера которых записаны в данном списке.

A3, B4/ C5: ошибка записи (сектор)

СТОРОЖЕВОЕ УСЛОВИЕ

является обычным булевским выражением и предназначено для синхронизации отдельных нитей потока управления.

Сторожевое условие записывается в квадратных скобках и может быть опущено, если оно отсутствует у данного сообщения.

Семантика сторожевого условия обеспечивает передачу сообщения только в том случае, если это условие принимает значение «истина».

Пример записи сторожевых условий без номеров предшествующих сообщений:

`[(x>=0)&(x<=255)] 1.2: отобразить_на_экране_цвет(x);`

`[количество цифр номера = 7] 3.1: набрать_телефонный_номер();`

ВЫРАЖЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ

это разделенный точками список отдельных термов последовательностей, после которого записывается двоеточие:

<Терм последовательности'. '><Терм последовательности'. '>':'

Каждый из термов - **отдельный уровень процедурной вложенности** в форме законченной итерации. Наиболее верхний уровень соответствует самому левому терму последовательности. Если все потоки управления параллельные, то вложенность отсутствует. Каждый из термов последовательности имеет следующий синтаксис:

[Целое число | Имя] [Символ рекуррентности]

Целое число указывает на **порядковый номер сообщения** в процедурной последовательности верхнего уровня. Сообщения, номера которых отличаются на единицу, следуют подряд один за другим.

Имя используется для спецификации параллельных нитей управления. Сообщения, которые отличаются только именем, являются параллельными на этом уровне вложенности. На одном уровне вложенности все потоки управления эквивалентны в смысле приоритета передачи сообщений.

Символ рекуррентности используется для указания условного или итеративного выполнения:

'* '[' Предложение-итерация ']' для записи итеративного выполнения соответствующего выражения. Итерация представляет последовательность сообщений одного уровня вложенности. Предложение-итерация может быть опущено, если условия итерации никак не специфицируются. Наиболее часто предложение-итерация записывается на некотором псевдокоде или языке программирования. В языке UML формат записи этого предложения не определен.

"*[/:=/..n]", означает последовательную передачу сообщения с параметром /, который изменяется от 1 до некоторого целого числа n с шагом 1;

'[' Предложение-условие У ']' для записи ветвления. Это условие представляет такое сообщение, передача которого по данной ветви возможна только при истинности этого условия. Чаще всего предложение-условие записывают на некотором псевдокоде или языке программирования, поскольку в языке UML формат записи этого предложения не определен.

[x>y] означает, что сообщение по некоторой ветви будет передано только в том случае, если значение x больше значения y.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

представляется в форме списка имен значений, возвращаемых по окончании коммуникации или взаимодействия в полной итерации данной процедурной последовательности. Эти идентификаторы могут выступать в качестве аргументов в последующих сообщениях. Если сообщение не возвращает никакого значения, то ни значение, ни оператор присваивания на диаграмме кооперации не указываются.

1.2.3: p:= найти_документ (спецификация_документа)

означает передачу вложенного сообщения с запросом поиска в базе данных нужного документа по его спецификации, источнику сообщения должен быть возвращен найденный документ.

ИМЯ СООБЩЕНИЯ

записанное в сигнатуре после возвращаемого значения, означает **ИМЯ события, которое инициируется объектом-получателем сообщения после его приема.**

Наиболее часто таким событием является вызов операции объекта.

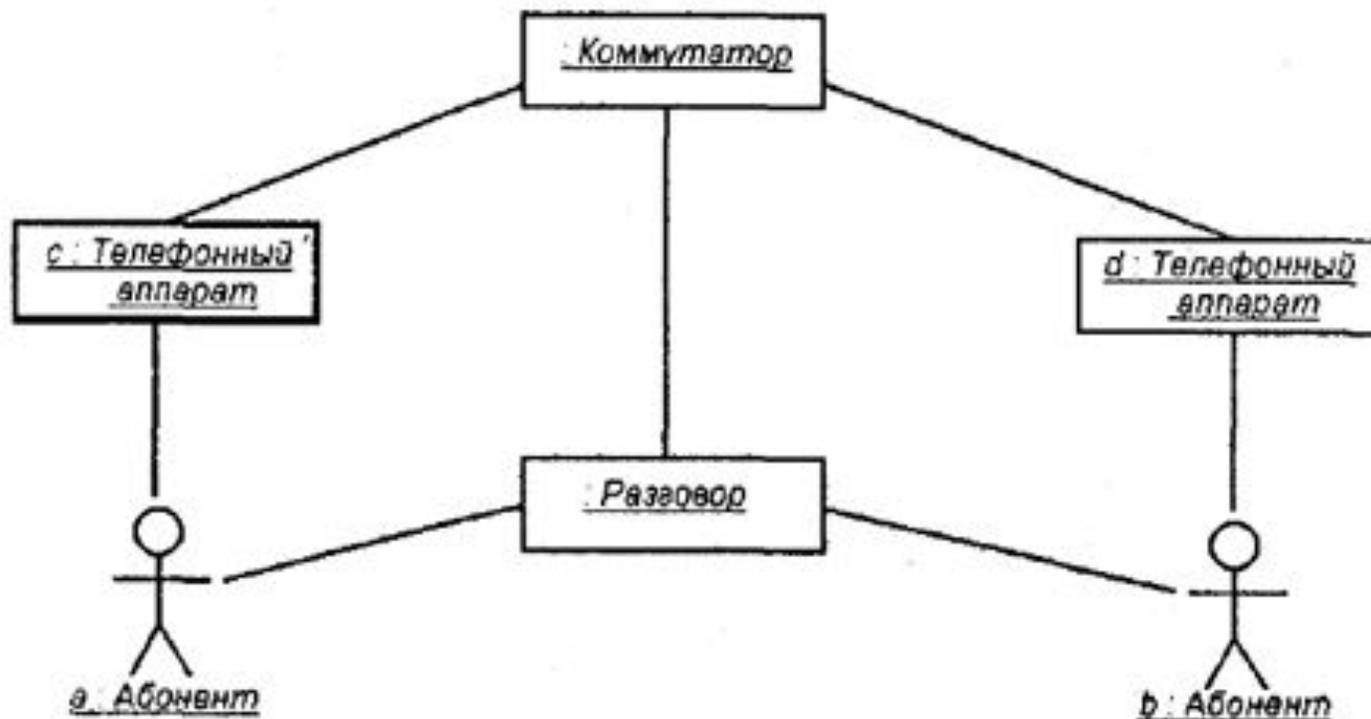
СПИСОК АРГУМЕНТОВ

представляет собой разделенные запятыми и заключенные в круглые скобки действительные параметры той операции, вызов которой инициируется данным сообщением. Список аргументов может быть пустым, однако скобки все равно записываются. Для записи аргументов также может быть использован некоторый псевдокод или язык программирования.

1.2.3: `p:= найти_документ (спецификация_документа)`

аргумент `найти_документ` является именем **сообщения**, а **спецификация_документа** - **списком аргументов**, состоящим из единственного действительного параметра операции. При этом имя сообщения означает обращение к операции `найти_документ`, которая должна быть определена в соответствующем классе объекта-получателя.

НАЧАЛЬНЫЙ ФРАГМЕНТ ДИАГРАММЫ КООПЕРАЦИИ



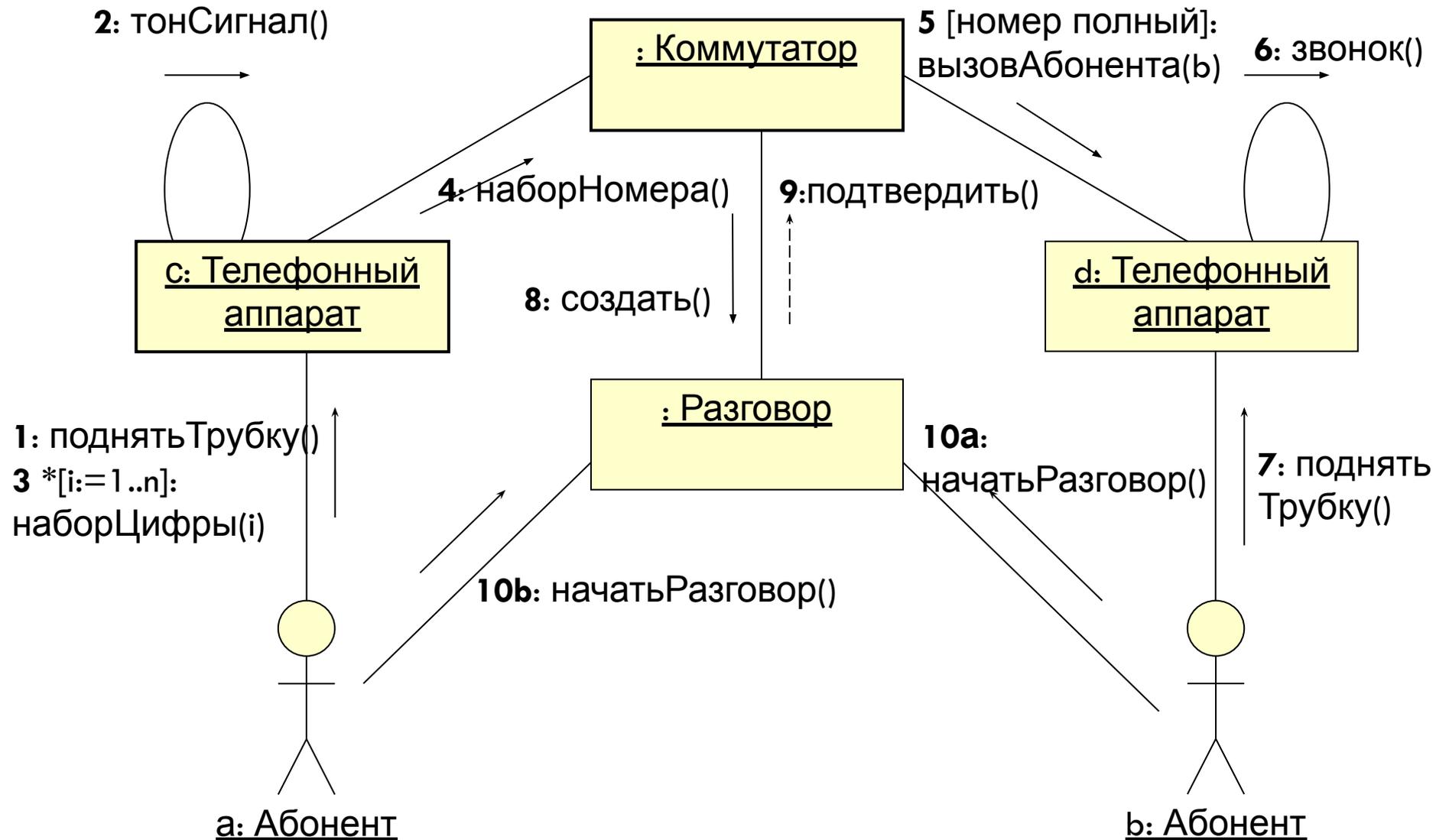
ДОПОЛНЕННЫЙ СТЕРЕОТИПАМИ РОЛЕЙ СВЯЗЕЙ, ИМЕНАМИ АССОЦИАЦИЙ И ПОМЕЧЕННЫМ ЗНАЧЕНИЕМ ОБЪЕКТА

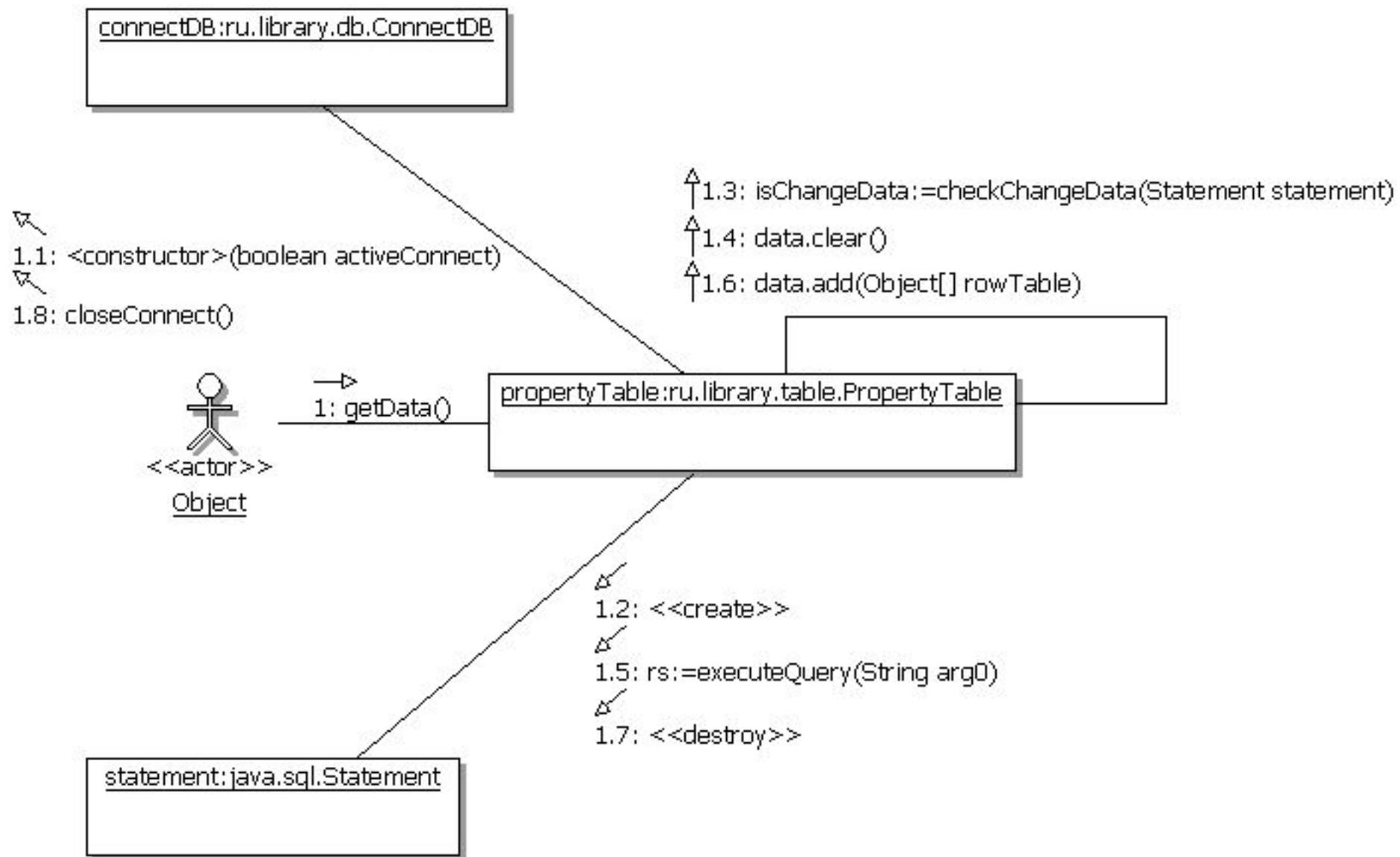


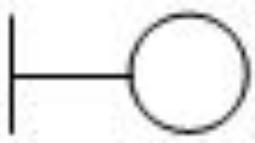
ОКОНЧАТЕЛЬНЫЙ ВАРИАНТ ДИАГРАММЫ КООПЕРАЦИИ

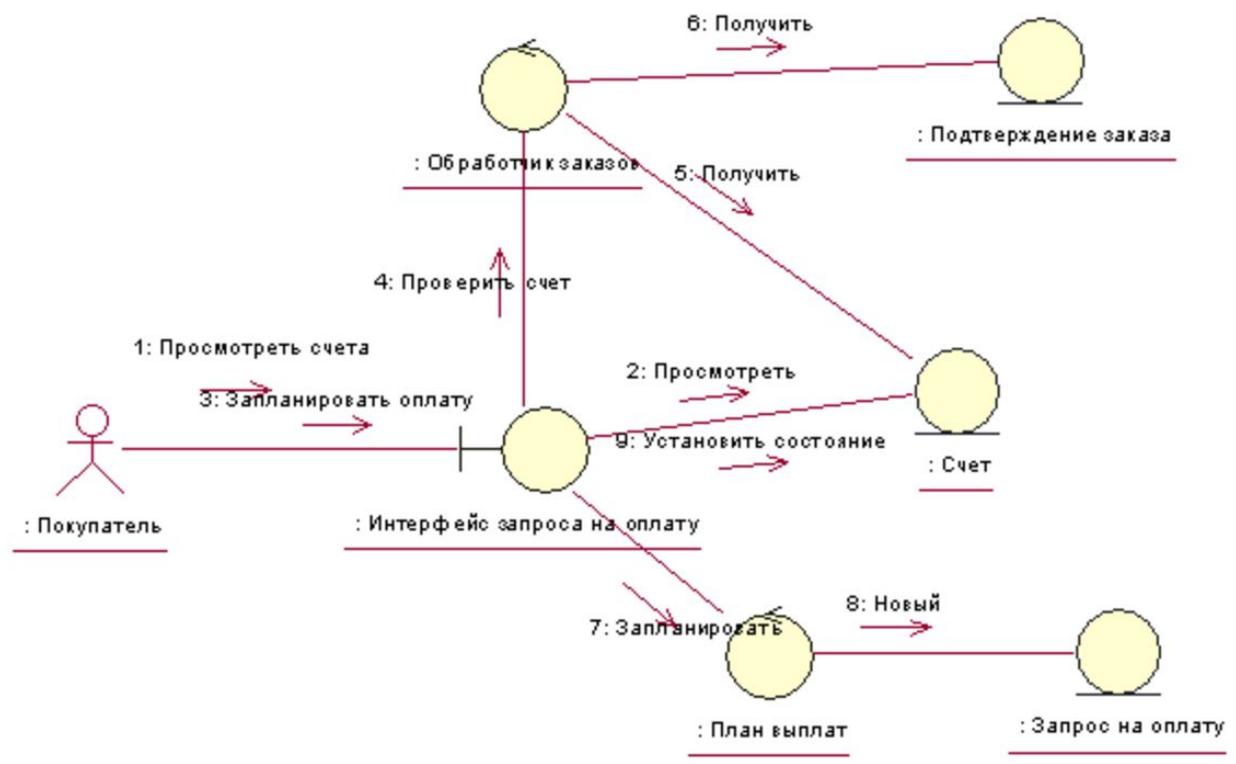


Пример диаграммы кооперации





Варианты отображения	Граничный класс	Управляющий класс	Класс сущности
Графический стереотип	 <p>Диалоговое окно «Нормативы»</p>	 <p>Расчет Vдоп</p>	 <p>План</p>
Стандартное обозначение со строкой-стереотипом	<div data-bbox="766 1150 1141 1315" style="border: 1px solid black; padding: 5px; text-align: center;"> <p>«boundary» Диалоговое окно «Нормативы»</p> </div>	<div data-bbox="1304 1150 1679 1315" style="border: 1px solid black; padding: 5px; text-align: center;"> <p>«control» Расчет Vдоп</p> </div>	<div data-bbox="1778 1150 2153 1315" style="border: 1px solid black; padding: 5px; text-align: center;"> <p>«entity» План</p> </div>





http://www.telenir.net/uchebniki/samouchitel_uml/index.php