

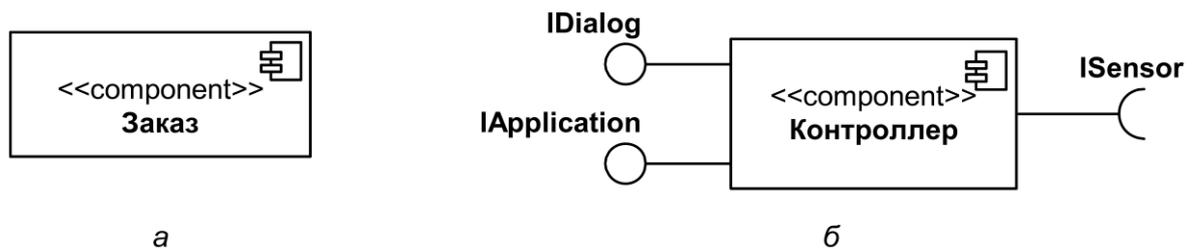
## 1.1. Диаграмма компонентов (component diagram)

Диаграмма компонентов, в отличие от ранее рассмотренных диаграмм, описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Во многих средах разработки модуль или компонент соответствует файлу. Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции исходных текстов программ. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

Диаграмма компонентов разрабатывается для следующих целей:

- визуализация общей структуры исходного кода программной системы;
- спецификация исполнимого варианта программной системы;
- обеспечение многократного использования отдельных фрагментов программного кода;
- представление концептуальной и физической схем баз данных.

Компонент (component) — элемент модели, представляющий некоторую модульную часть системы с инкапсулированным содержимым, спецификация которого является взаимозаменяемой в его окружении.



**Рис. 11.1.** Примеры изображения: а — простого компонента; б — компонента с интерфейсами

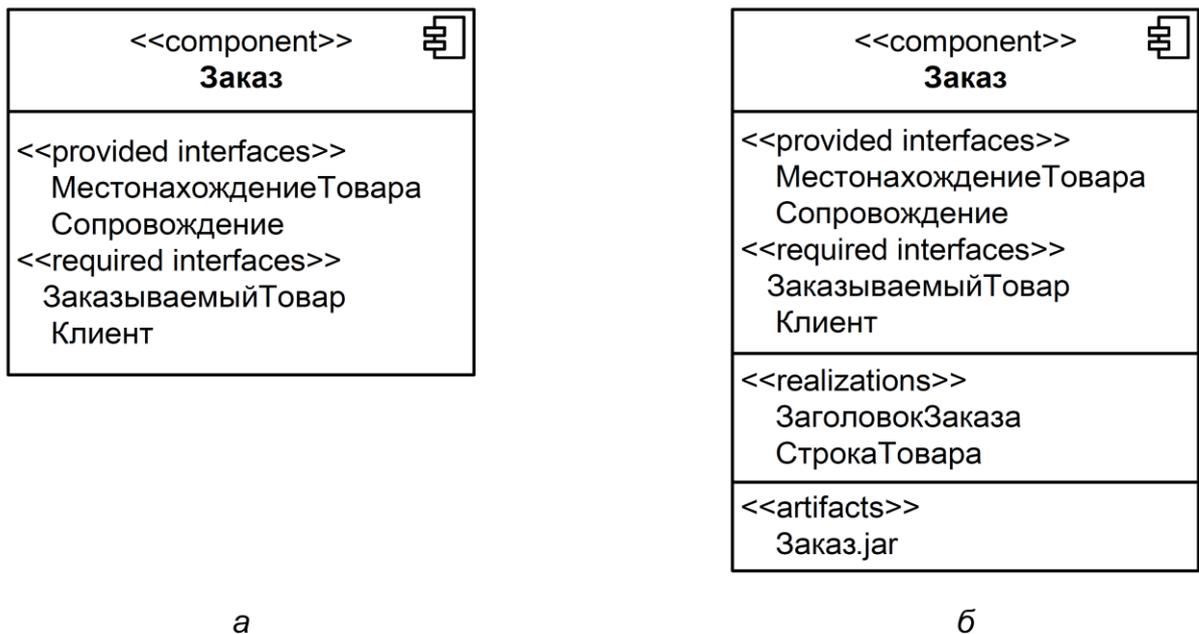


Рис. 11.2. Примеры изображения компонента в нотации:  
 а — черного ящика; б — белого ящика

### 1.1.1. Интерфейс

Предоставляемый интерфейс (provided interface) — интерфейс, который компонент предлагает для своего окружения.

Требуемый интерфейс (required interface) — интерфейс, который необходим компоненту от своего окружения для выполнения заявленной функциональности, контракта или поведения.

Предоставляемые и требуемые интерфейсы в языке UML 2.0 изображаются с использованием специальной нотации, которая получила название "шара и шарнира" или "леденца на палочке". Эта нотация позволяет компактно изобразить графическое соединение компонентов и удовлетворяет требованию масштабирования сложных систем. Предоставляемые интерфейсы компонента обозначаются линией с шаром, а требуемые интерфейсы — линией с гнездом.

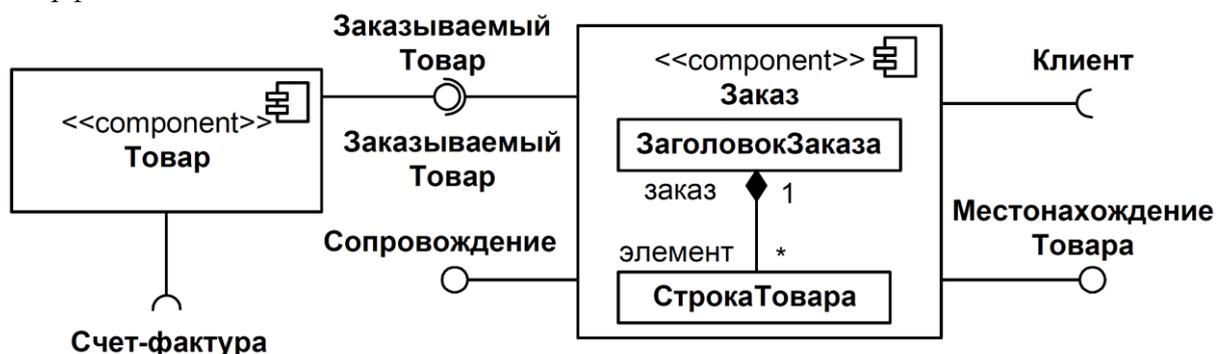


Рис. 11.3. Пример изображения компонентов с интерфейсами в нотации "леденца на палочке"

Порт определяет различимую точку взаимодействия между компонентом и окружающей его средой или между компонентом и его внутренними частями. Изображаются порты так же, как и на других диаграммах структуры — в виде небольших квадратов, расположенных на границе символа компонента.

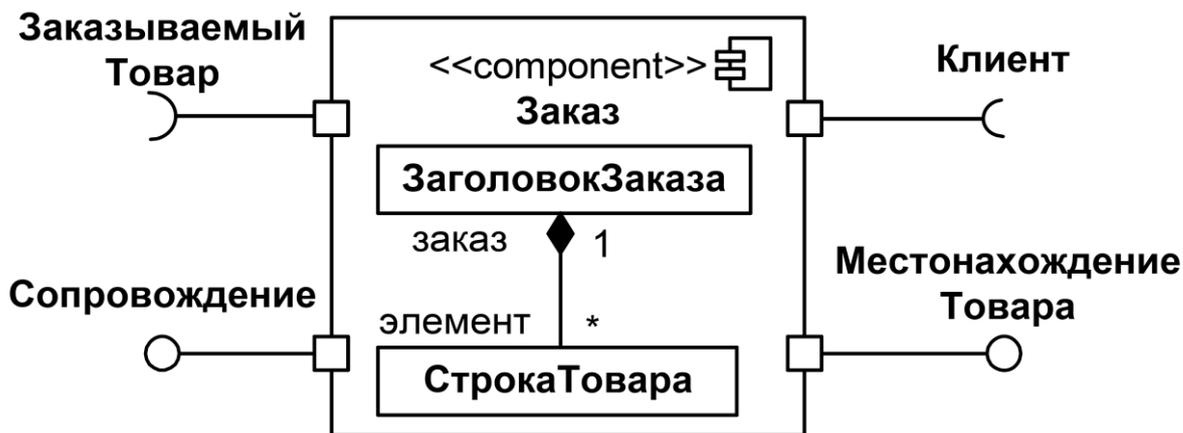


Рис. 11.5. Примеры изображения портов компонентов

### 1.1.2. Соединители

Соединитель представляет собой некоторую связь, которая обеспечивает коммуникацию между двумя и более компонентами. Различают два вида соединителей: собирающие соединители и делегирующие соединители.

Собирающий соединитель (assembly connector) — соединитель, который связывает два компонента в контексте предоставляемых и требуемых сервисов.

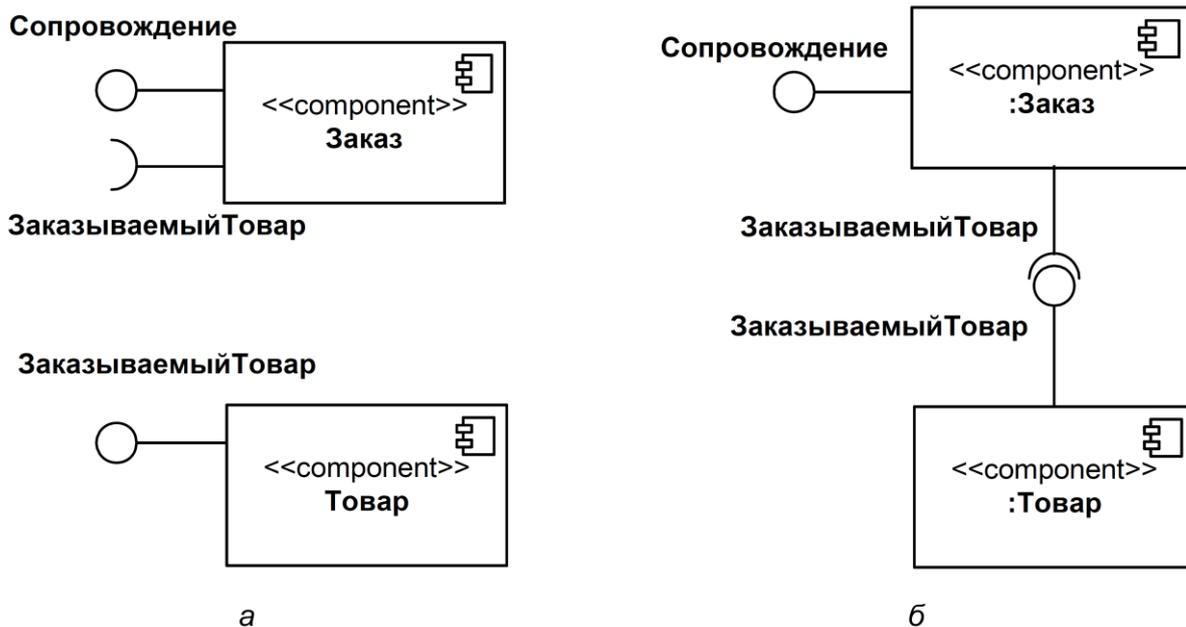


Рис. 11.6. Собирающий соединитель: а — в библиотеке компонентов; б — в конкретной сборке

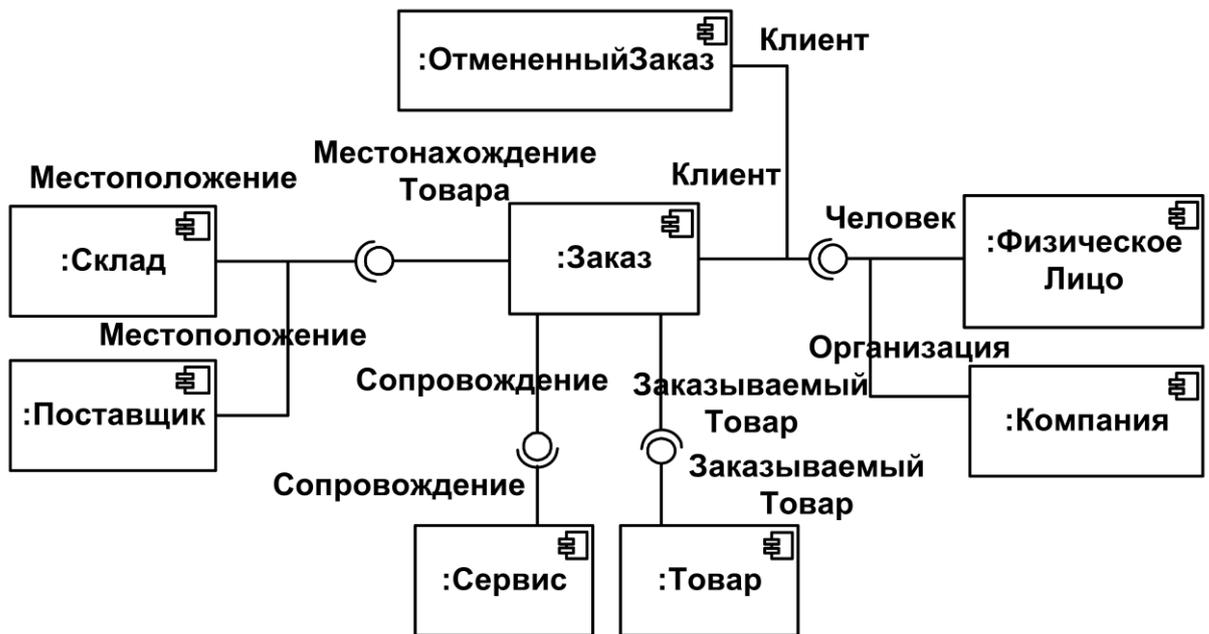


Рис. 11.7. Пример диаграммы компонентов с собирающими соединителями для одинаковых интерфейсов

Делегирующий соединитель (delegation connector) — соединитель, который связывает внешний контракт компонента с реализацией этого поведения внутренними частями этого компонента.

В общем случае делегирующий соединитель выполняет одну из следующих задач:

- передача сообщений или сигналов, поступающих в порт компонента из вне, для обработки в некоторую внутреннюю часть компонента или другой порт;
- передача сообщений или сигналов, поступающих из некоторой внутренней части компонента, для обработки во внешний порт компонента.

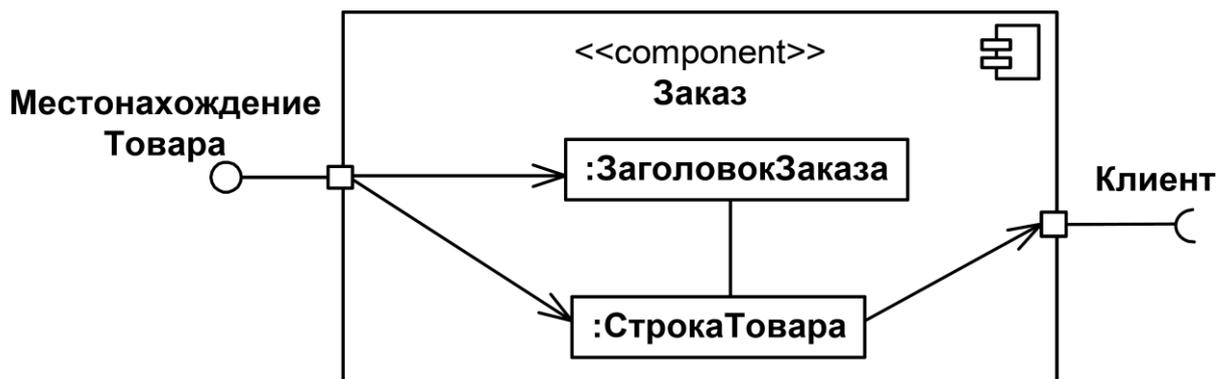
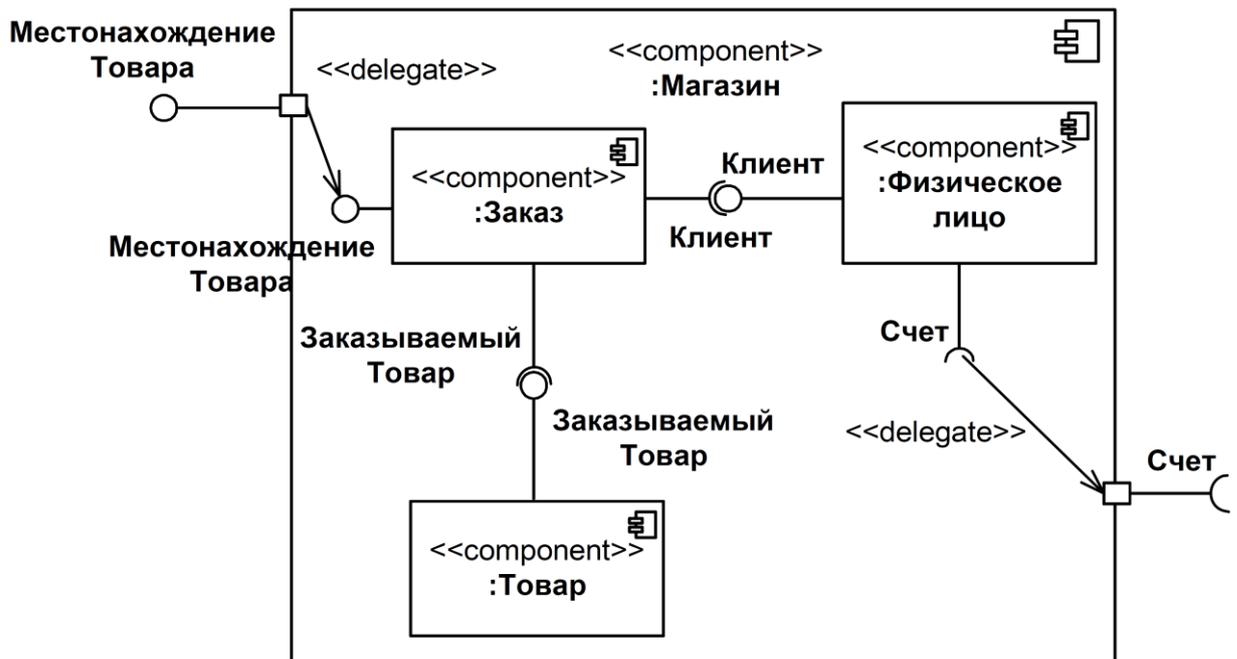


Рис. 11.8. Примеры изображения делегирующих соединителей



**Рис. 11.9.** Пример внутренней структуры экземпляра компонента, который содержит другие компоненты как свои части

При использовании делегирующих соединителей в модели необходимо помнить о следующих ограничениях.

- Делегирующий соединитель может быть определен только между интерфейсами или портами одного и того же вида. А именно, или между предоставляемым интерфейсом и портом, или между требуемым интерфейсом и портом.
- Если делегирующий соединитель определяется между требуемым интерфейсом (портом) и некоторой внутренней частью, то эта внутренняя часть должна иметь отношение реализации с интерфейсом (портом) этого типа.
- Если делегирующий соединитель определяется между интерфейсом (портом) в качестве источника и интерфейсом (портом) в качестве цели, то интерфейс в качестве цели должен поддерживать сигнатуру, совместимую с подмножеством операций интерфейса (порта) источника.

### 1.1.3. Отношения

На диаграмме компонентов также могут быть представлены отношения зависимости между компонентами. Это представление часто называют обзорным, поскольку на нем не изображаются интерфейсы и порты.

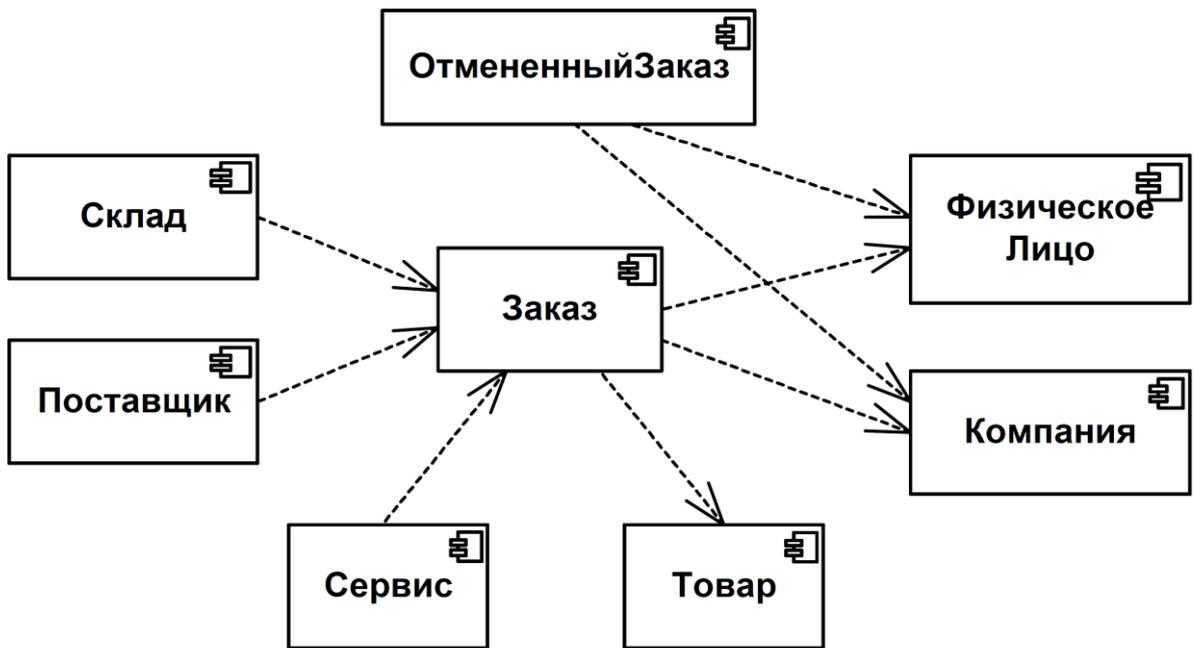


Рис. 11.10. Пример отношений зависимости между компонентом

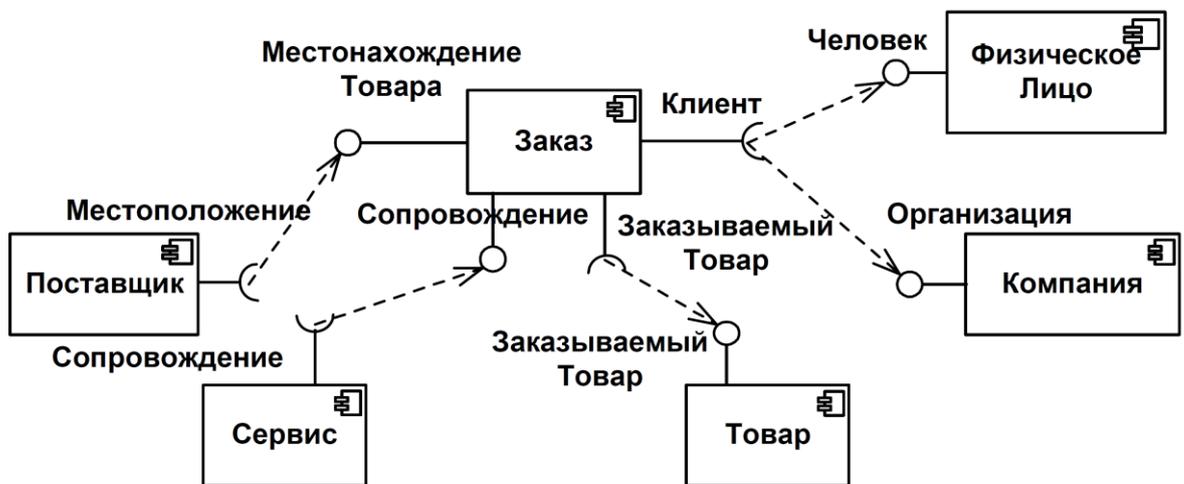
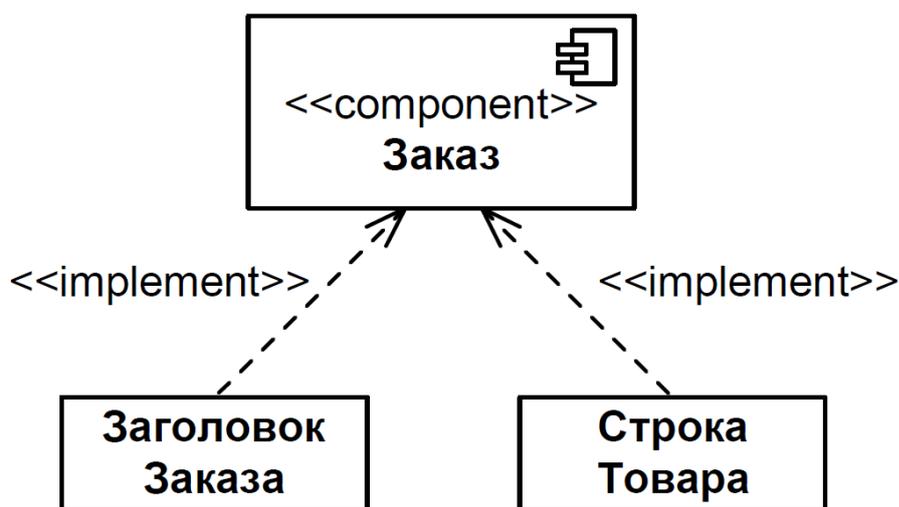


Рис. 11.11. Отношения зависимости на диаграмме компонентов с интерфейсами

Реализация (realization) — специализация отношения зависимости для связи компонентов с классификаторами, которые реализуют функциональность этого компонента.



**Рис. 11.12.** Пример отношения реализации компонента

#### 1.1.4. Компонент

Компонент, как элемент модели, может иметь различную физическую реализацию. Для более точной спецификации формы реализации компонента можно использовать текстовый стереотип, который указывается выше строки с именем компонента.

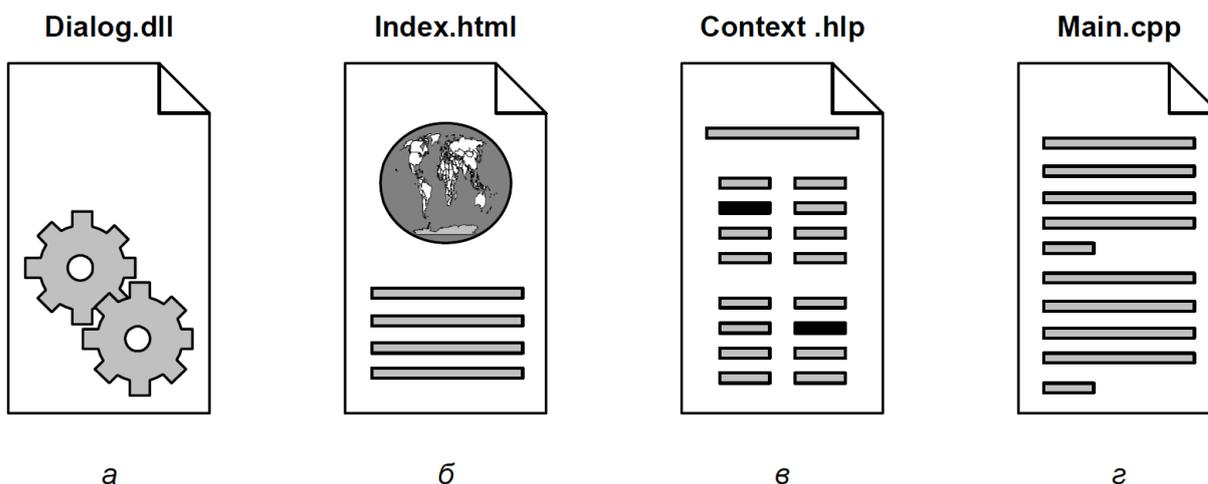
**Таблица 11.1.** Основные текстовые стереотипы компонентов

Ключевое слово	Перевод	Назначение
file	Файл	Определяет наиболее общую разновидность компонента, который представляется в виде произвольного физического файла
executable	Исполнимый	Определяет разновидность компонента-файла, который является исполнимым файлом и может выполняться на некоторой компьютерной платформе
document	Документ	Определяет разновидность компонента-файла, который представляется в форме документа произвольного содержания, не являющегося исполнимым файлом или файлом с исходным текстом программы
library	Библиотека	Определяет разновидность компонента-файла, который представляется в форме динамической или статической библиотеки
source	Источник	Определяет разновидность компонента-файла, представляющего собой файл с исходным текстом программы, который после компиляции может быть преобразован в исполнимый файл
table	Таблица	Определяет разновидность компонента, который представляется в форме таблицы базы данных

Из графических стереотипов наибольшее распространение получили три группы символов.

Для первой группы компонентов используются следующие стереотипы:

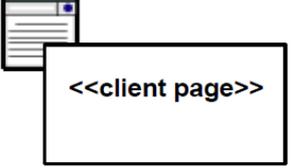
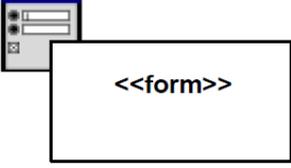
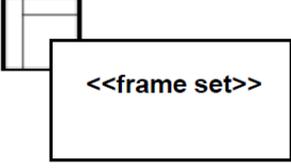
- динамически подключаемые библиотеки, как правило имеющие расширение DLL (рис. 11.13, а);
- Web-страницы на языке разметки гипертекста, как правило, имеющие расширение HTML или HTML (рис. 11.13, б);
- файлы справки, как правило, имеющие расширение HLP (рис. 11.13, в);
- файлы с исходными текстами программ, имеющие, например, расширения H и CPP для языка C++ (рис. 11.13, г).



**Рис. 11.13.** Изображение графических стереотипов компонентов Г. Буча:  
а — динамически подключаемых библиотек; б — Web-страниц; в — файлов справки;  
г — файлов с исходными текстами программ

Вторая группа стереотипов

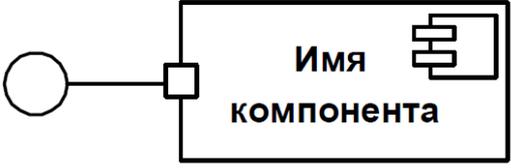
Таблица 11.2. Графические стереотипы компонентов Дж. Коналлена

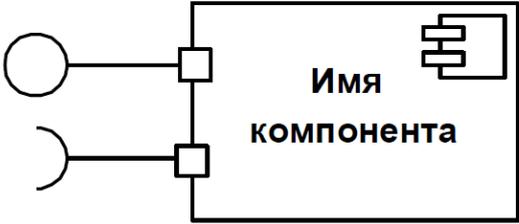
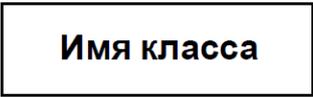
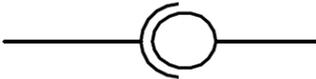
Графический символ	Назначение
	<p>Серверная страница представляет Web-страницу, содержащую выполняемые сервером сценарии. Эти сценарии могут взаимодействовать с серверными ресурсами, такими как базы данных, бизнес-логика и внешние системы. Операции реализуемых компонентов классов являются функциями сценария, а их атрибуты — переменными, видимыми в пределах этой страницы</p>
	<p>Клиентская страница представляет Web-страницу в формате HTML, а также данные, элементы интерфейса и даже бизнес-логику. Клиентские страницы отображаются клиентскими браузерами и могут содержать сценарии, которые интерпретируются браузером. Операции клиентской страницы могут соответствовать функциям, содержащимся в дескрипторах сценария страницы. Атрибутам клиентской страницы соответствуют объявленные в дескрипторах сценария переменные, которые доступны любой функции в пределах этой страницы</p>
	<p>Форма является набором полей ввода и представляет собой часть клиентской страницы. Форма преобразуется непосредственно в дескриптор HTML <code>&lt;form&gt;</code>. Атрибуты формы могут представлять поля ввода, текстовые поля, переключатели, флажки, скрытые поля формы HTML. С формой не связано никаких операций, поскольку их нельзя в ней инкапсулировать. Любые операции взаимодействия с формой являются свойствами содержащей ее страницы</p>
	<p>Набор фреймов представляет собой контейнер, состоящий из нескольких Web-страниц. Прямоугольная область просмотра делится на несколько фреймов. Каждый фрейм может быть связан с одним объектом со стереотипом <code>&lt;&lt;target&gt;&gt;</code>, однако это необязательно. Содержимым фрейма может быть Web-страница или другой фрейм. Набор фреймов преобразуется непосредственно в набор фреймов Web-страницы и дескриптор HTML <code>&lt;frame&gt;</code>. Поскольку на практике набор фреймов является клиентской страницей, этот компонент также может включать классы с операциями и атрибутами, которые могут активизироваться лишь теми браузерами, которые не отображают фреймы</p>

Графический символ	Назначение
 <div data-bbox="308 293 555 416" style="border: 1px solid black; padding: 5px; width: fit-content;"> <p data-bbox="363 322 499 349">&lt;&lt;target&gt;&gt;</p> </div>	<p data-bbox="619 237 1455 539">Цель представляет собой именованную область окна браузера, в которой могут отображаться Web-страницы. Имя цели соответствует имени целевого объекта. Обычно целью является один из фреймов набора. Однако целью может быть и новое окно браузера. Для цели может быть задано место назначения, где будет отображена новая Web-страница. Имя цели должно быть уникальным для каждого клиента системы. Таким образом, для каждого клиента может существовать лишь один экземпляр цели</p>
 <div data-bbox="308 595 555 719" style="border: 1px solid black; padding: 5px; width: fit-content;"> <p data-bbox="339 624 523 651">&lt;&lt;web-page&gt;&gt;</p> </div>	<p data-bbox="619 551 1455 920">Этот компонент представляет Web-страницу, которую браузер может запрашивать по ее имени. Этот компонент при необходимости может содержать клиентские или серверные сценарии. Обычно Web-страницы являются текстовыми файлами, доступ к которым можно получить через Web-сервер. Однако они могут быть также компилируемыми модулями, загружаемыми и запускаемыми Web-сервером. В любом случае при доступе к такой странице, хранящейся в файле или исполняемой Web-сервером, она генерирует документ в формате HTML, который отправляется в ответ на запрос браузера</p>
 <div data-bbox="308 965 555 1088" style="border: 1px solid black; padding: 5px; width: fit-content;"> <p data-bbox="339 981 523 1043">&lt;&lt;java server page&gt;&gt;</p> </div>	<p data-bbox="619 931 1455 1099">Этот компонент представляет Web-страницы, реализующие код JSP серверной части приложения. Данный стереотип применим лишь к приложениям, в которых используется технология Java Server Pages</p>
 <div data-bbox="308 1178 555 1301" style="border: 1px solid black; padding: 5px; width: fit-content;"> <p data-bbox="355 1216 507 1243">&lt;&lt;servlet&gt;&gt;</p> </div>	<p data-bbox="619 1111 1455 1319">Этот компонент представляет сервлет Java. Стереотип применим лишь к приложениям, поддерживающим сервлеты компании Sun</p>

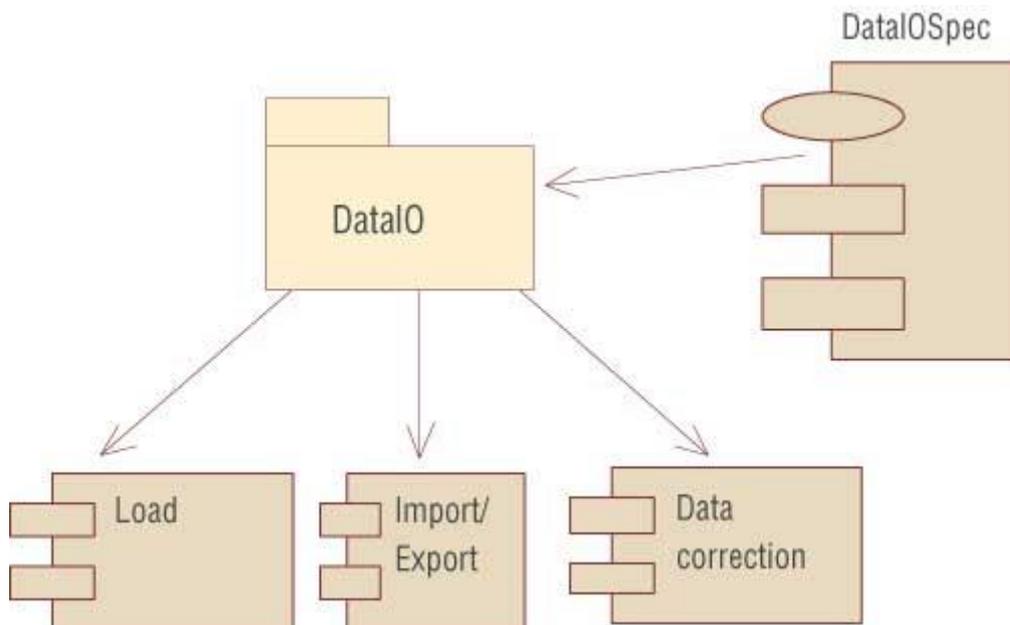
### 1.1.5. Свободная таблица

**Таблица 11.3.** Графические узлы и пути на диаграмме

Тип графического элемента	Нотация
Компонент (component) с текстовым стереотипом	
Компонент (component) с пиктограммой стереотипа	
Компонент с предоставляемым интерфейсом (provided interface)	
Компонент имеет порт (port) с предоставляемым интерфейсом	
Компонент с требуемым интерфейсом (required interface)	
Компонент имеет порт (port) с требуемым интерфейсом	

Тип графического элемента	Нотация
Компонент имеет несколько портов (ports) с предоставляемыми и требуемыми интерфейсами	
Класс (class)	
Часть (part)	
Собирающий соединитель (assembly connector)	

### 1.1.6. Примеры диаграмм



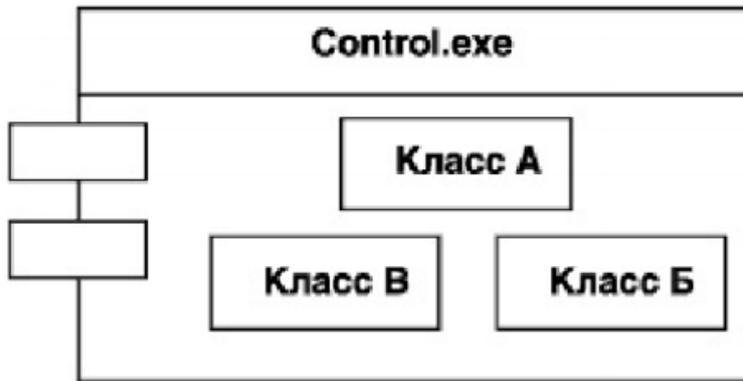


Рис. 12.7. Графическое изображение компонента с информацией о реализуемых им классах.

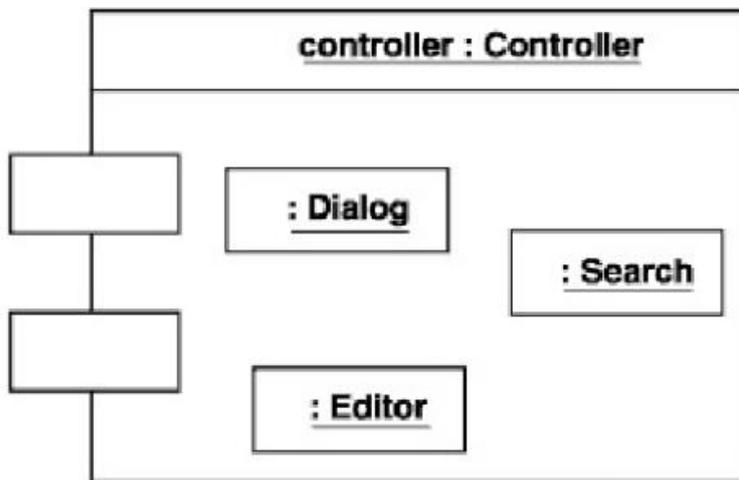


Рис. 12.8. Графическое изображение компонента-экземпляра, реализующего отдельные объекты.

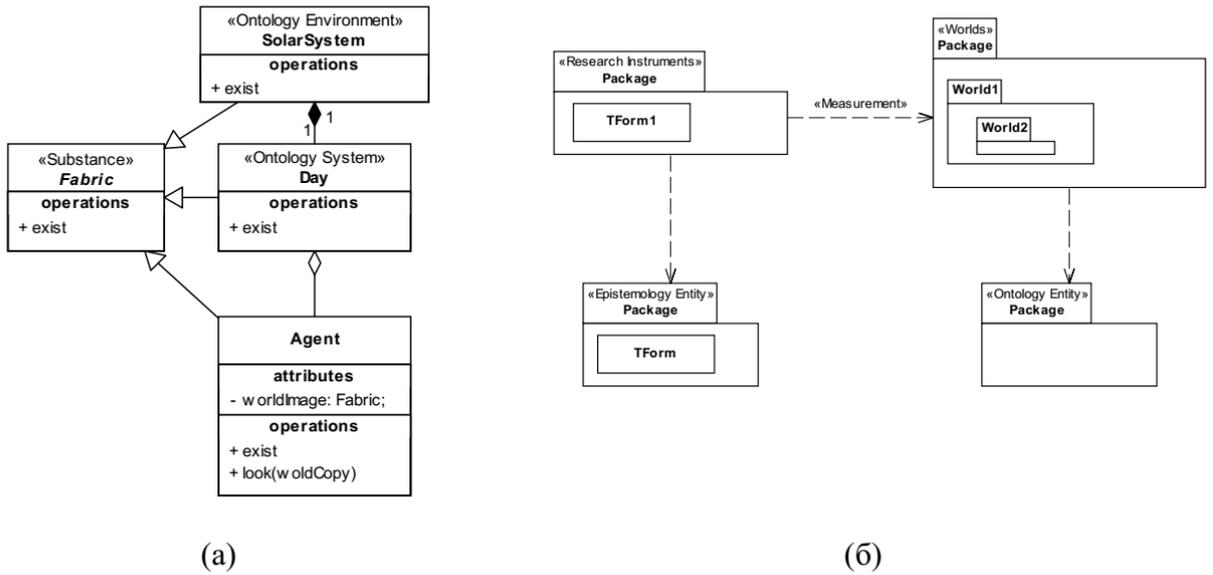


Рис. 20. Диаграмма классов (а) и архитектура (б) для вложенных моделей Sun

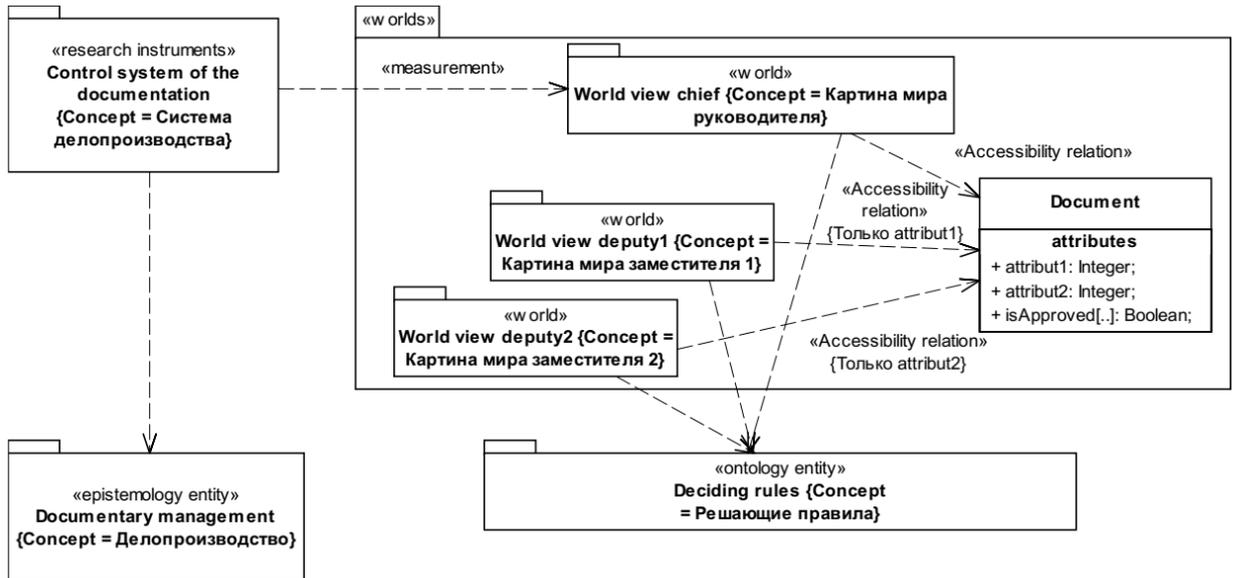


Рис. 22. Архитектура пакета Analysis System (Research Analysis Model) для задачи двух заместителей