Учебный курс

Язык UML в анализе и проектировании программных систем и бизнес-процессов

Лекция 7

Диаграмма компонентов языка UML 2

Диаграмма компонентов

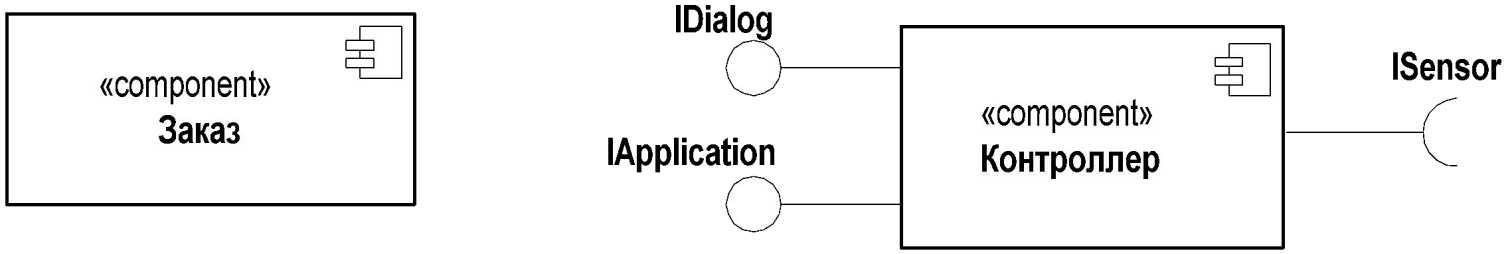
- – диаграмма физического уровня, которая служит для представления программных компонентов и зависимостей между ними.
- Диаграмма компонентов разрабатывается для следующих целей:
  - Визуализация общей структуры исходного кода программной системы.
  - Спецификация исполнимого варианта программной системы.
  - Обеспечение многократного использования отдельных фрагментов программного кода.
  - Представление концептуальной и физической схем баз данных.

Компонент *(component)*

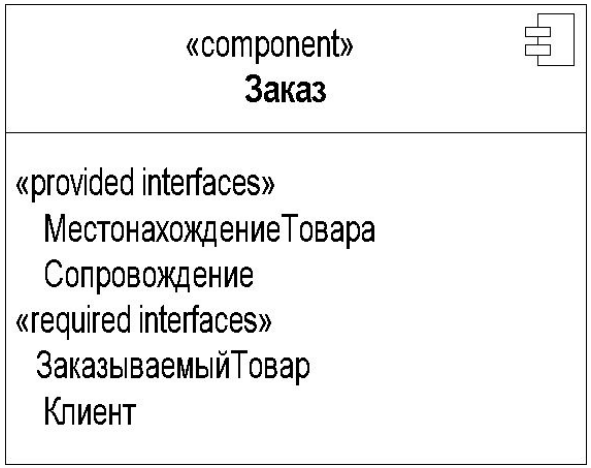
- – элемент модели, представляющий некоторую модульную часть системы с инкапсулированным содержимым, спецификация которого является взаимозаменяемой в его окружении.
- Имя экземпляра компонента записывается аналогично имени линии жизни на диаграммах взаимодействия в следующем формате (БНФ):

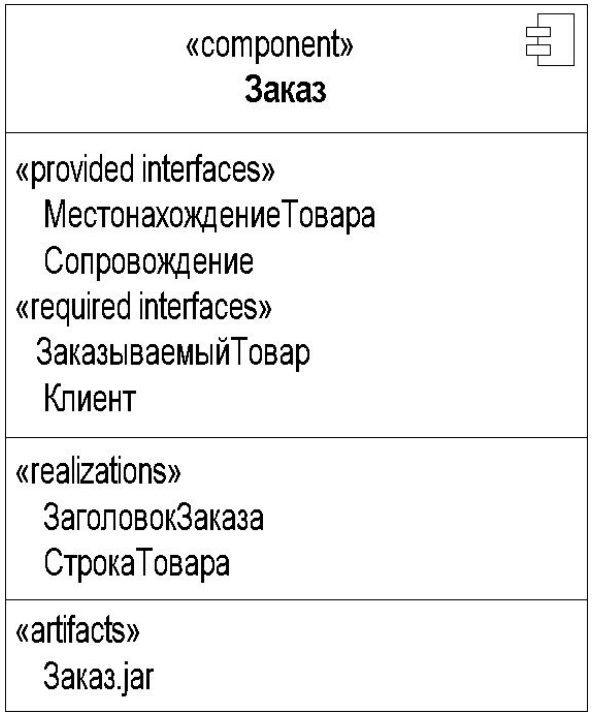
  <*имя-экземпляра-компонента*>::=[<*собственное-имя-компонента*>] [*':'*<*имя-типа*>],
- при этом *собственное имя компонента* записывается со строчной буквы, а в качестве имени экземпляра компонента должен присутствовать хотя бы один терм.

Примеры изображения простого компонента и компонента с интерфейсами



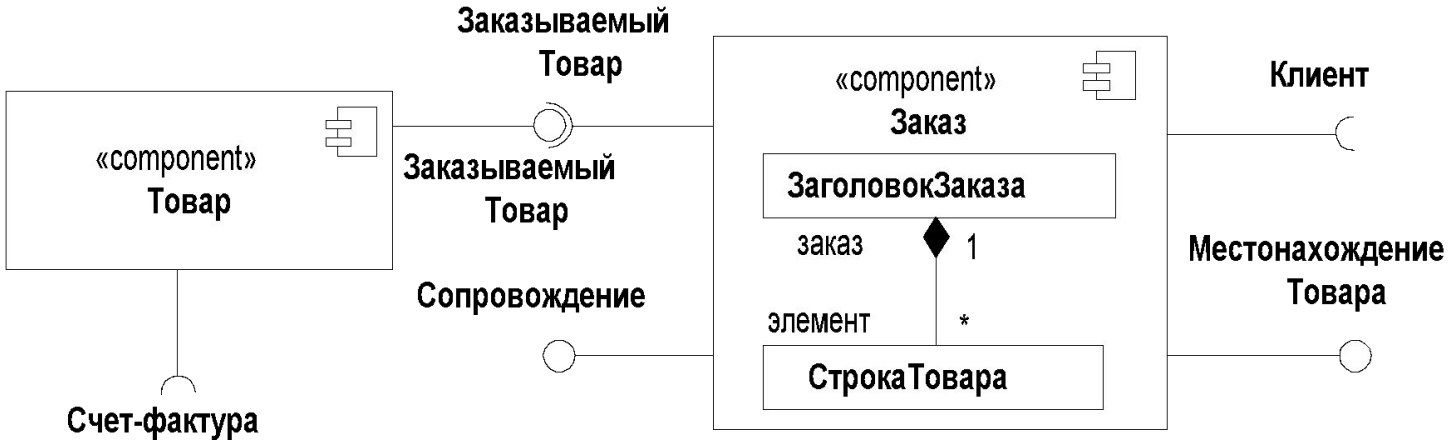
Примеры изображения компонента в нотации черного и белого ящика



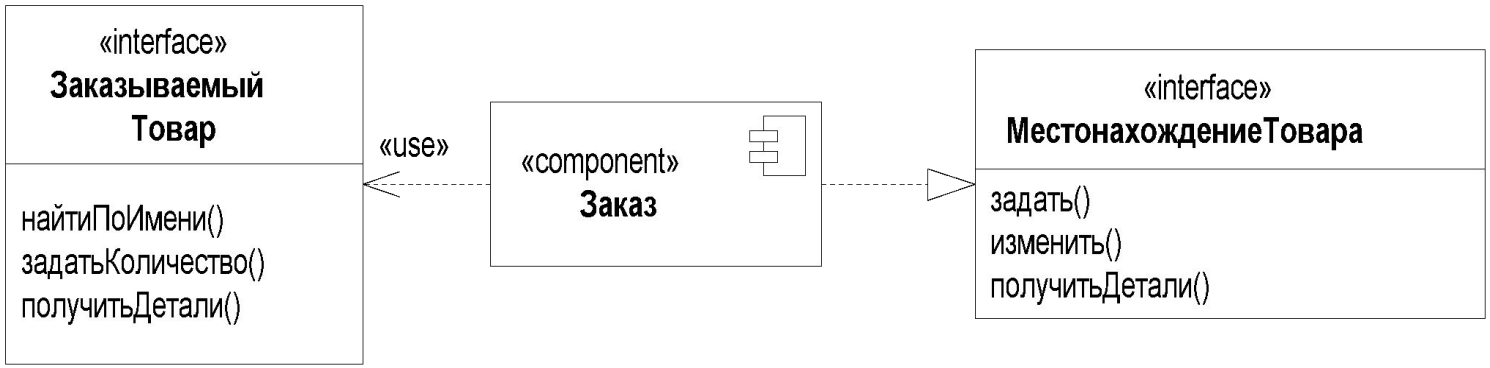


Интерфейсы

- *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.

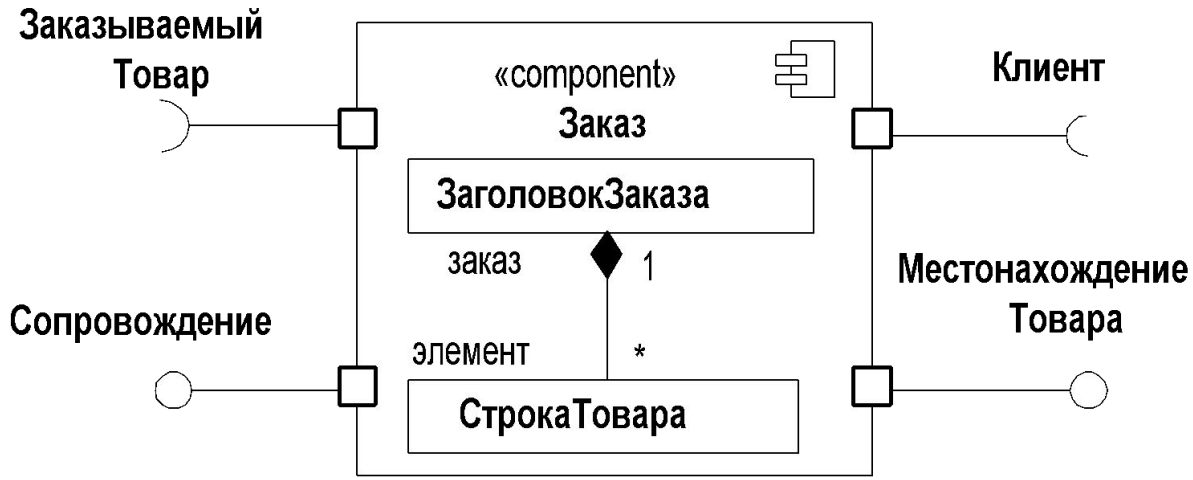


Представление интерфейсов в форме символа классификатора с отношениями зависимости и реализации



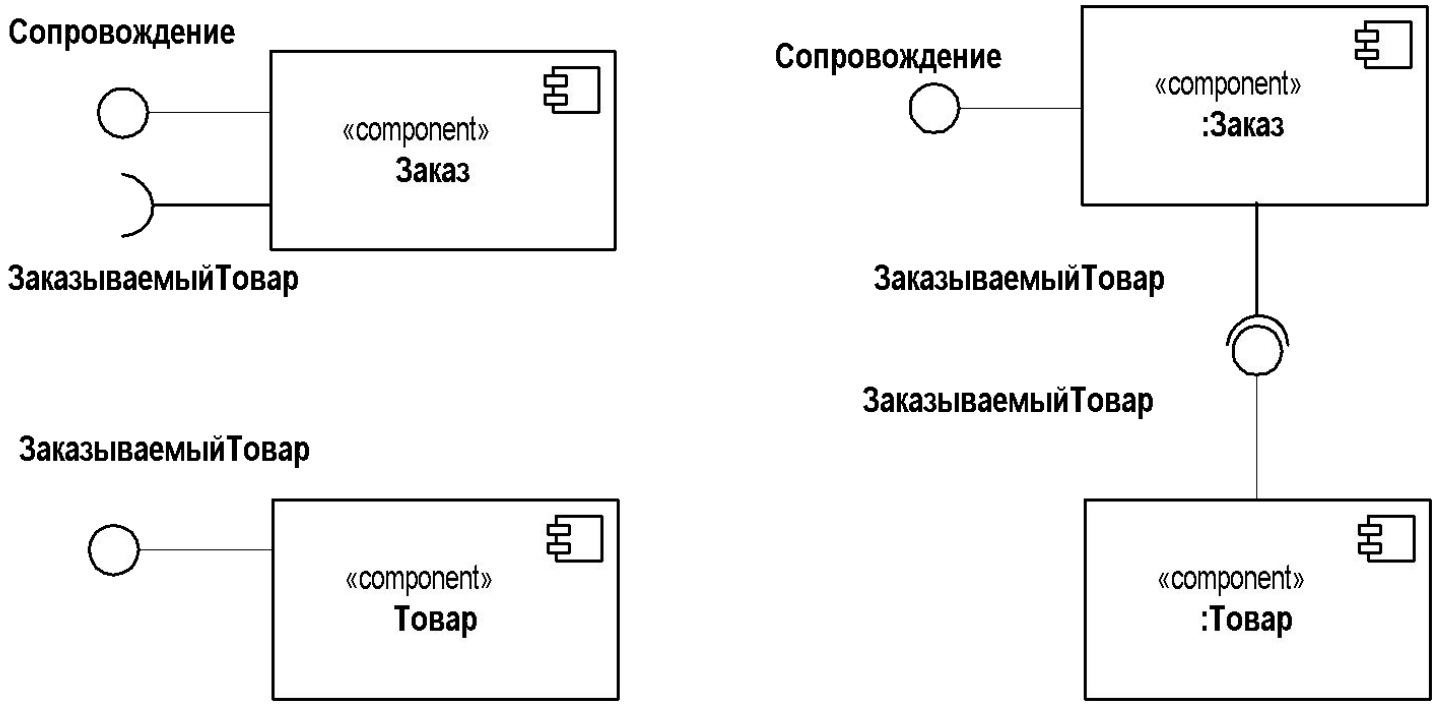
Порты

- Порт определяет различимую точку взаимодействия между компонентом и окружающей его средой или между компонентом и его внутренними частями
- Наличие имени у порта не является обязательным
- При отсутствии имени порта его тип ассоциируется с типом интерфейса, с которым связан порт.



Собирающий соединитель *(assembly connector)*

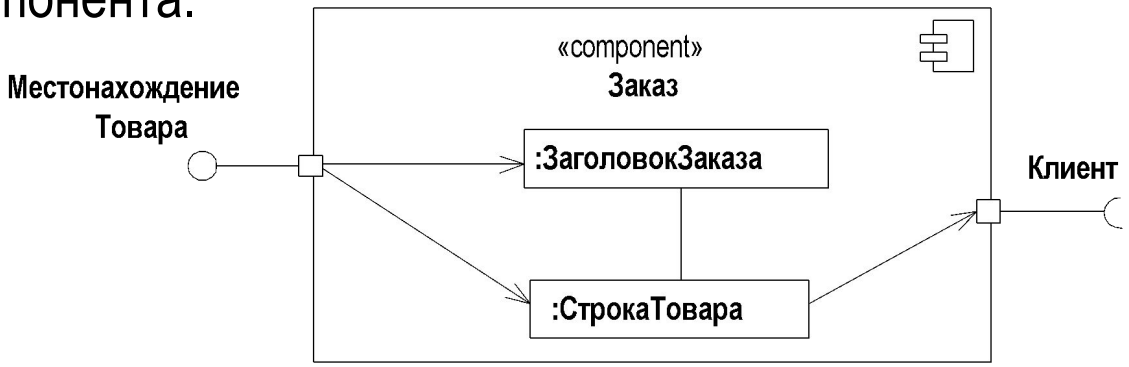
- – соединитель, который связывает два компонента в контексте предоставляемый и требуемых сервисов.



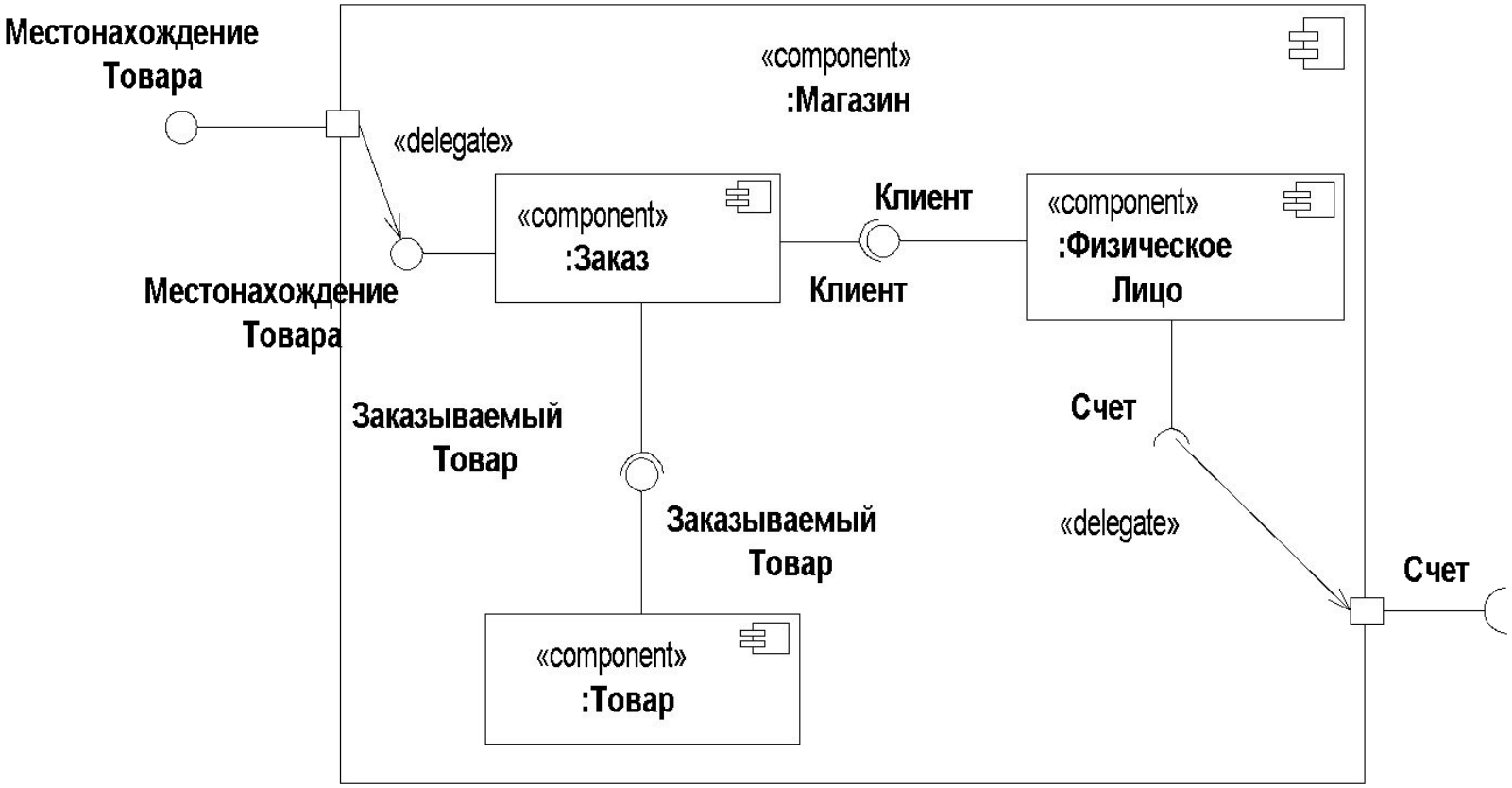
Пример диаграммы компонентов с собирающими соединителями для одинаковых интерфейсов

Делегирующий соединитель *(delegation connector)*

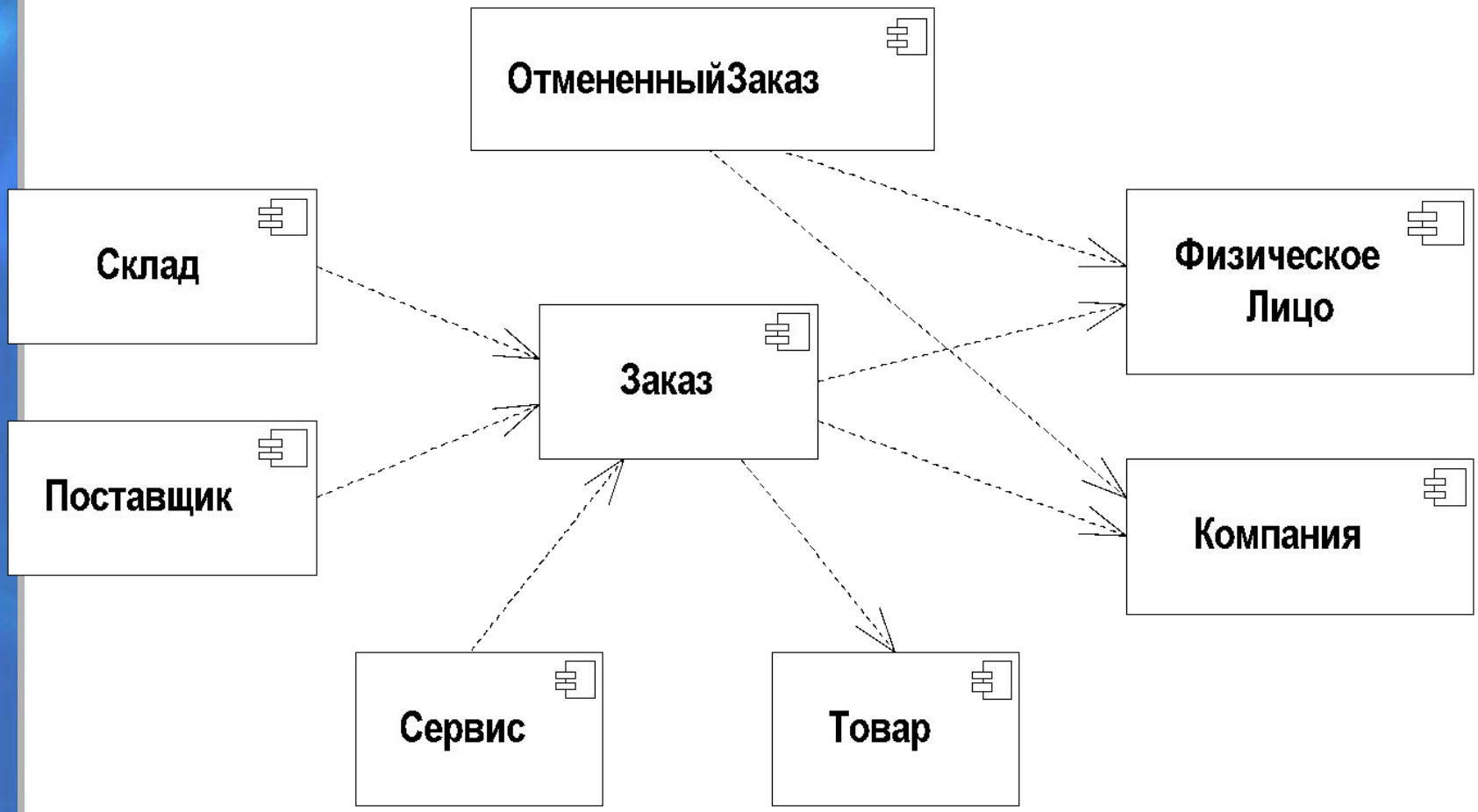
- – соединитель, который связывает внешний контракт компонента с реализацией этого поведения внутренними частями этого компонента.
- Делегирующий соединитель выполняет одну из следующих задач:
  - Передача сообщений или сигналов, поступающих в порт компонента извне, для обработки в некоторую внутреннюю часть компонента или другой порт.
  - Передача сообщений или сигналов, поступающих из некоторой внутренней части компонента, для обработки во внешний порт компонента.



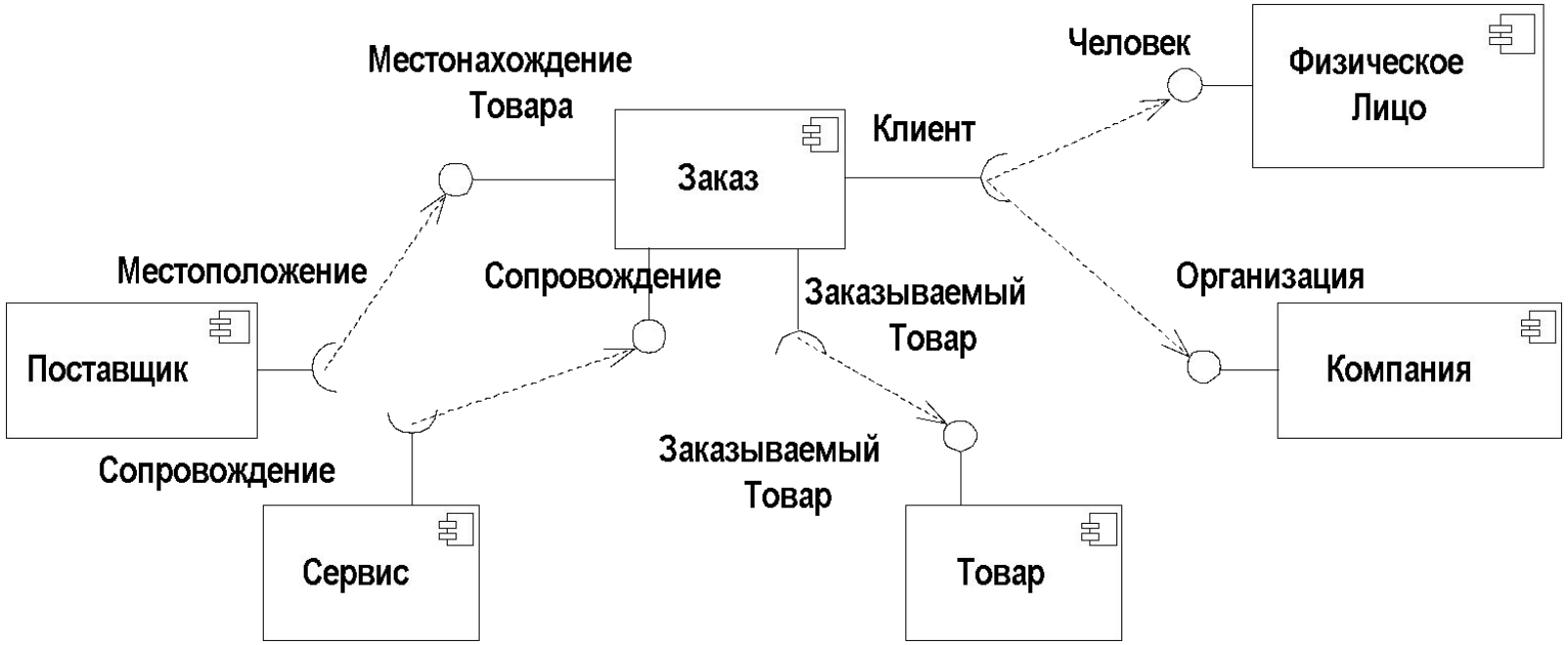
Пример внутренней структуры экземпляра компонента



Пример отношений зависимости между компонентом

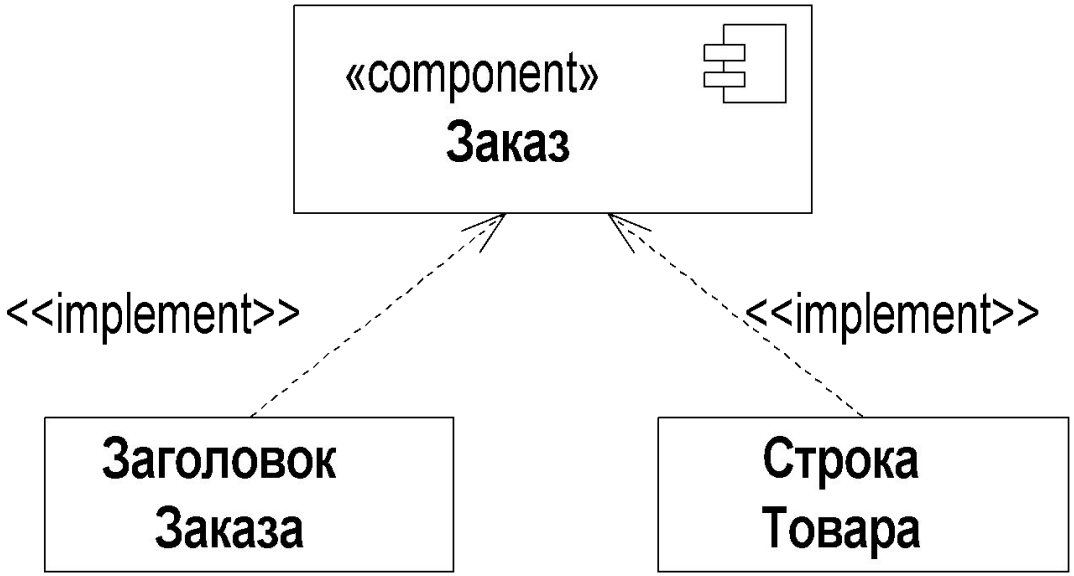


Отношения зависимости на диаграмме компонентов с интерфейсами

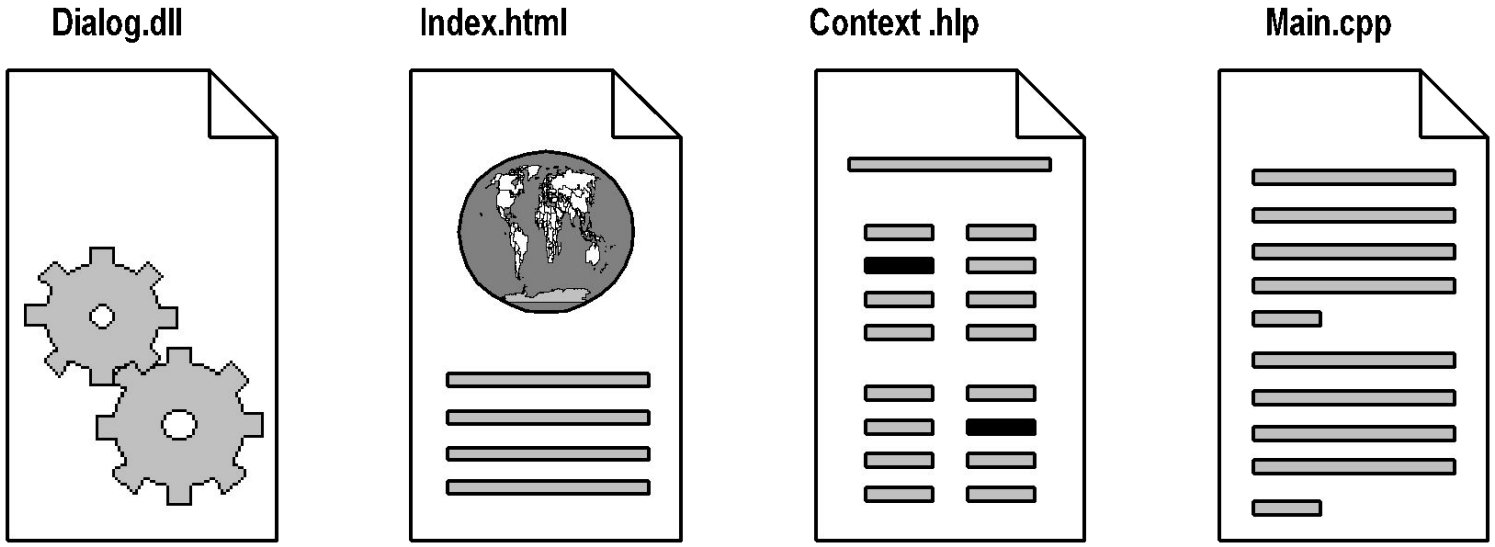


Реализация *(realization)*

- – специализация отношения зависимости для связи компонентов с классификаторами, которые реализуют функциональность этого компонента
- Реализация компонента может быть дополнительно помечена стереотипом «implement»



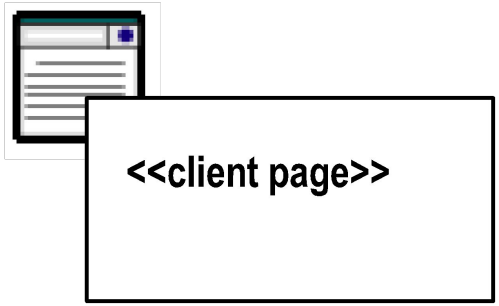
Изображение графических стереотипов компонентов Г.Буча



Графические стереотипы компонентов Дж. Коналлена

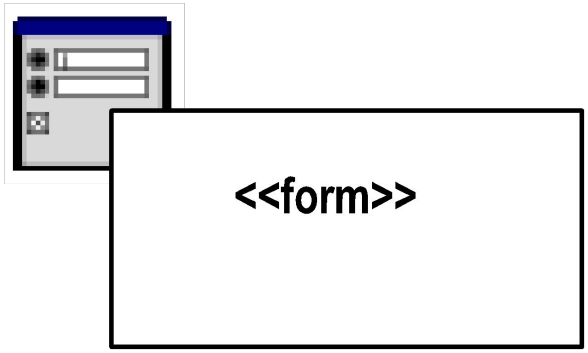
- Серверная страница представляет Web-страницу, содержащую выполняемые сервером сценарии.
- Эти сценарии могут взаимодействовать с серверными ресурсами, такими как базы данных, бизнес-логика и внешние системы.
- Операции реализуемых компонент классов являются функциями сценария, а их атрибуты — переменными, видимыми в пределах этой страницы.

Клиентская страница



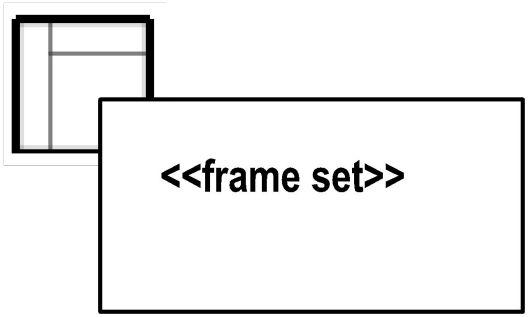
- Представляет Web-страницу в формате HTML, а также данные, элементы интерфейса и даже бизнес-логику.
- Клиентские страницы отображаются клиентскими броузерами и могут содержать сценарии, которые интерпретируются броузером.
- Операции клиентской страницы могут соответствовать функциям, содержащимся в дескрипторах сценария страницы.
- Атрибутам клиентской страницы соответствуют объявленные в дескрипторах сценария переменные, которые доступны любой функции в пределах этой страницы.

Форма



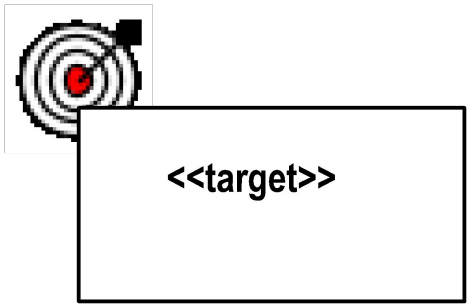
- Является набором полей ввода и представляет собой часть клиентской страницы.
- Форма преобразуется непосредственно в дескриптор HTML <form>.
- Атрибуты формы могут представлять поля ввода, текстовые поля, переключатели, флажки, скрытые поля формы HTML.
- С формой не связано никаких операций, поскольку их нельзя в ней инкапсулировать.
- Любые операции взаимодействия с формой являются свойствами содержащей ее страницы.

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 «target», однако это необязательно.
- Содержимым фрейма может быть Web-страница или другой фрейм. Набор фреймов преобразуется непосредственно в набор фреймов Web-страницы и дескриптор HTML <frame>.

Цель



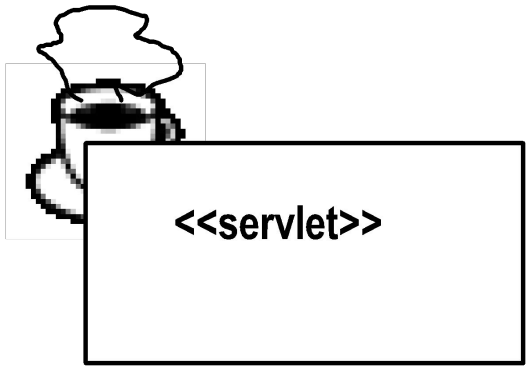
- Представляет собой именованную область окна броузера, в которой могут отображаться Web-страницы.
- Имя цели соответствует имени целевого объекта.
- Обычно целью является один из фреймов набора.
- Однако целью может быть и новое окно броузера. Для цели может быть задано место назначения, где будет отображена новая Web-страница.
- Имя цели должно быть уникальным для каждого клиента системы.

Web-страница

- Броузер может запрашивать Web-страницу 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 броузера.

JSP и сервлет

- Этот компонент представляет Web-страницы, реализующие код JSP серверной части при ложения. Этот стереотип применим лишь к приложениям, в которых используется технология Java Server Pages.
- Этот компонент представляет сервлет Java. Стереотип применим лишь к приложениям, поддерживающим сервлеты компании Sun.



Самостоятельное задание №8

- Выполнить текущее тестирование: вопросы 34-36
- Разработать диаграмму компонентов для ATM
  - Изобразить следующие компоненты: Главная программа, Программа обслуживания банкомата, Transaction, Устройства банкомата.
  - Интерфейс IATMBank
  - Поместить на диаграмму все классы, представленные на диаграмме классов
  - Изобразить отношения между ними