Научные методы в проектировании

Лекция 13

Топология информационных систем

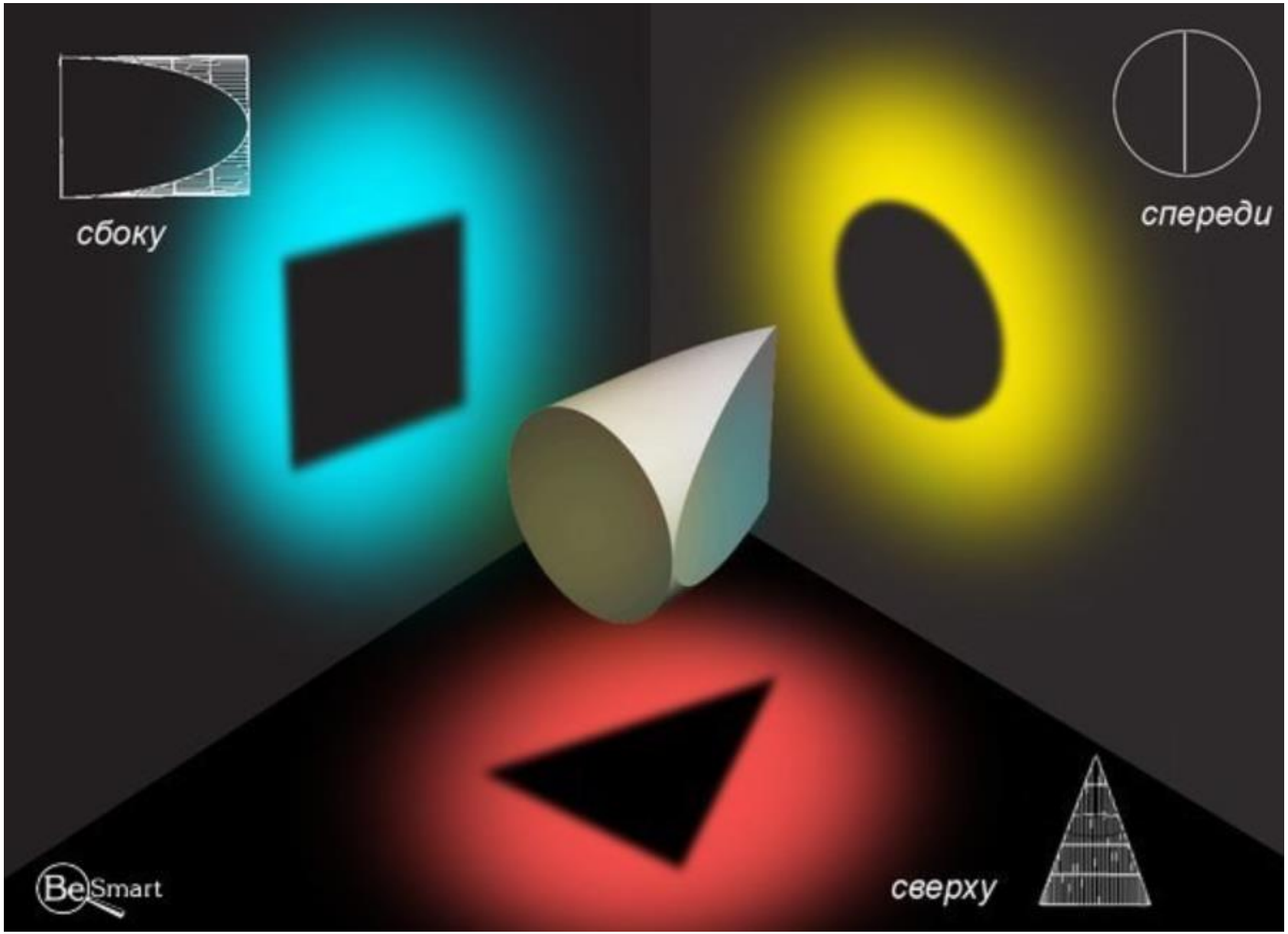
Овчинников П.Е.

МГТУ «СТАНКИН»,

ст.преподаватель кафедры ИС

10.12.2020

Точка зрения определяет все



Забавно

Топология в математике и сетях

Тополо́гия (от др.-греч. τόπος — место и λόγος — слово, учение) — раздел математики, изучающий:

- в самом общем виде — явление непрерывности;
- в частности — свойства пространств, которые остаются неизменными при непрерывных деформациях

В отличие от геометрии, в топологии не рассматриваются метрические свойства объектов (например, расстояние между парой точек). Например, с точки зрения топологии, кружка и бублик (полноторий) — неотличимы

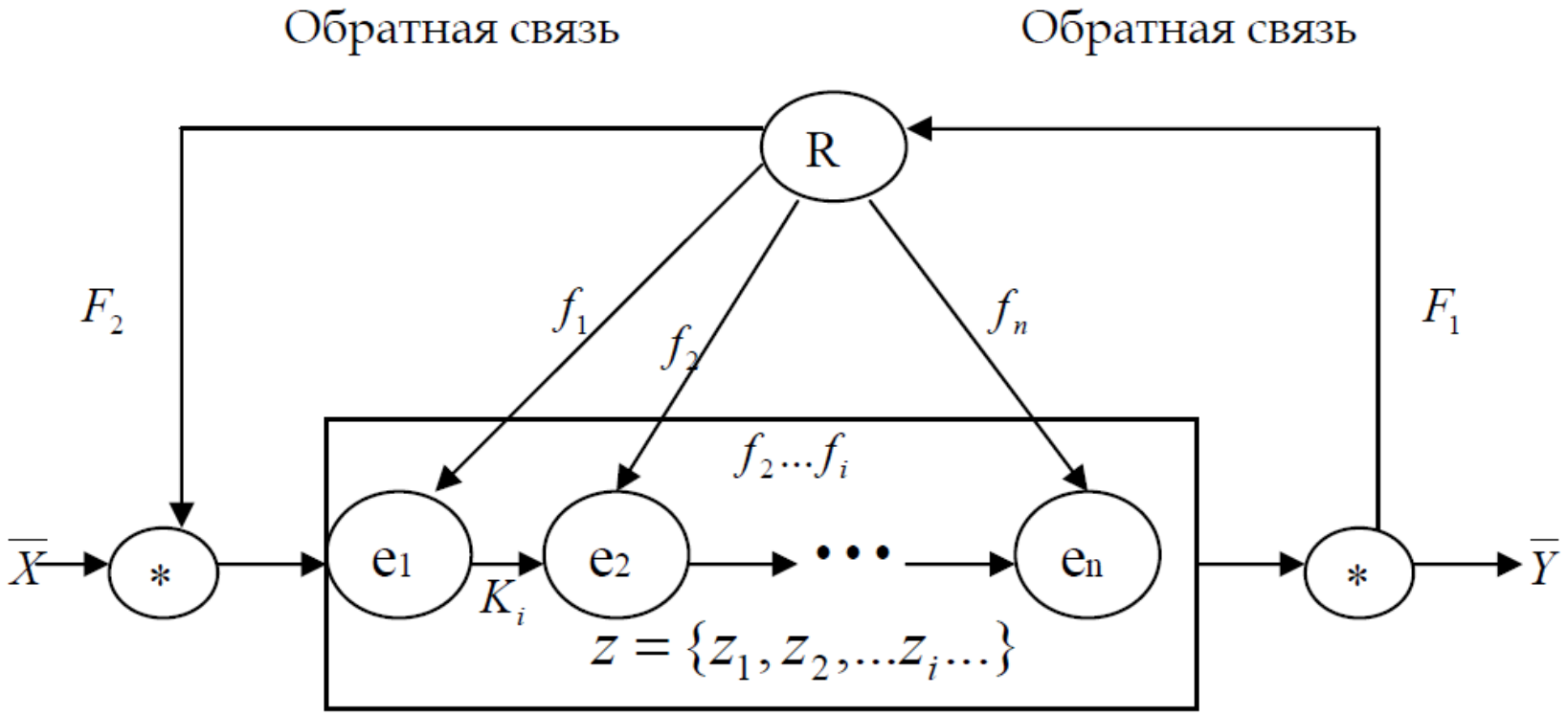
Сетевая тополо́гия — это конфигурация графа, вершинам которого соответствуют конечные узлы сети (компьютеры) и коммуникационное оборудование (маршрутизаторы), а рёбрам — физические или информационные связи между вершинами

Сетевая топология может быть

- **физической** — описывает реальное расположение и связи между узлами сети
- **логической** — описывает хождение сигнала в рамках физической топологии
- **информационной** — описывает направление потоков информации, передаваемых по сети
- **управления обменом** — это принцип передачи права на пользование сетью

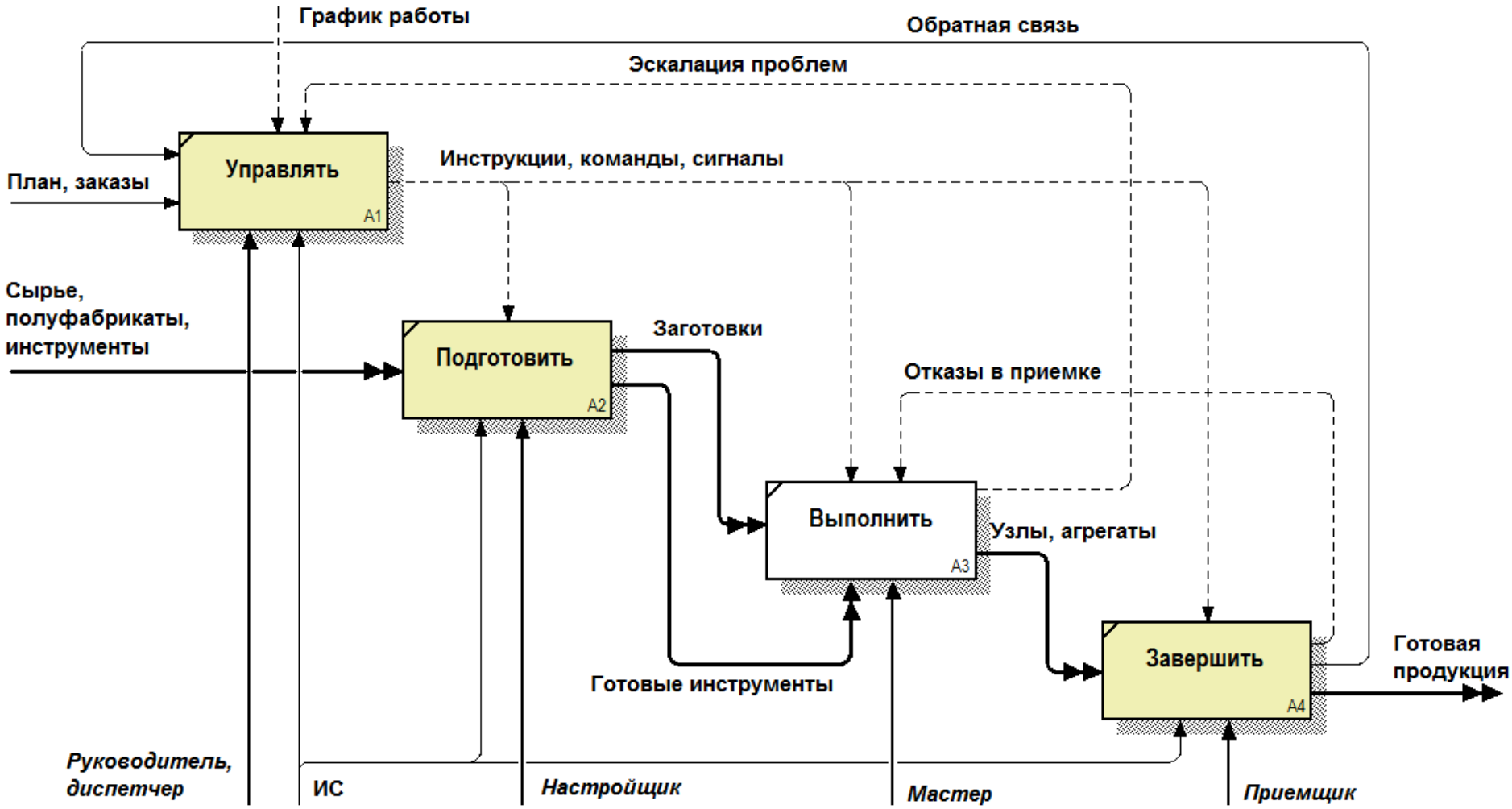
Топология в системах управления

Система ∑ – это конечная совокупность элементов (E) и некоторого регулирующего устройства (R), которое устанавливает связи между элементами (e_i) по преобразованию и управлению, управляет этими связями, создавая неделимую единицу функционирования. Топологически система представлена на рис. 1.



Данелян Т.Я.
Д177 ТЕОРИЯ СИСТЕМ И СИСТЕМНЫЙ АНАЛИЗ (ТСиСА): учебно-методический комплекс / Т.Я. Данелян. – М.: Изд. центр ЕАОИ, 2010. – 303 с.
ISBN 978-5-374-00324-6

Прямые и обратные связи в управлении



ООП: Классы и объекты

Класс (class)

категория вещей, которые имеют общие **атрибуты** и **операции**

Объект (object) -

- конкретная **материализация** абстракции;
- сущность с хорошо определенными **границами**, в которой **инкапсулированы состояние и поведение**;
- **экземпляр** класса

Объект **уникально идентифицируется** значениями атрибутов, определяющими его состояние в данный момент времени

Атрибут класса (class attribute)

именованное свойство класса, описывающее множество значений, которые могут принимать экземпляры этого свойства

Операция класса (class operation)

метод или функция, которая может быть выполнена экземпляром класса или интерфейсом

Объекты в программировании

Свойство объекта (property)

способ доступа к внутреннему состоянию объекта, имитирующий переменную некоторого типа.

Метод объекта (method) в объектно-ориентированном программировании это функция или процедура, принадлежащая какому-то классу или объекту

Как и процедура в процедурном программировании, метод состоит из некоторого количества операторов для выполнения какого-то действия и имеет набор входных аргументов

Инкапсуляция (англ. *encapsulation*, от лат. *in capsula*) в информатике упаковка **данных** и **функций** в единый компонент

В ООП инкапсуляция тесно связана с принципом абстракции данных (не путать с абстрактными типами данных, реализации которых предоставляют возможность инкапсуляции, но имеют иную природу).

Это, в частности, приводит к другому распространённому заблуждению — рассмотрению инкапсуляции неотрывно от сокрытия. В частности, в сообществе C++ или Java принято рассматривать инкапсуляцию без сокрытия как неполноценную

Неявные классы: модуль, переменная, функция

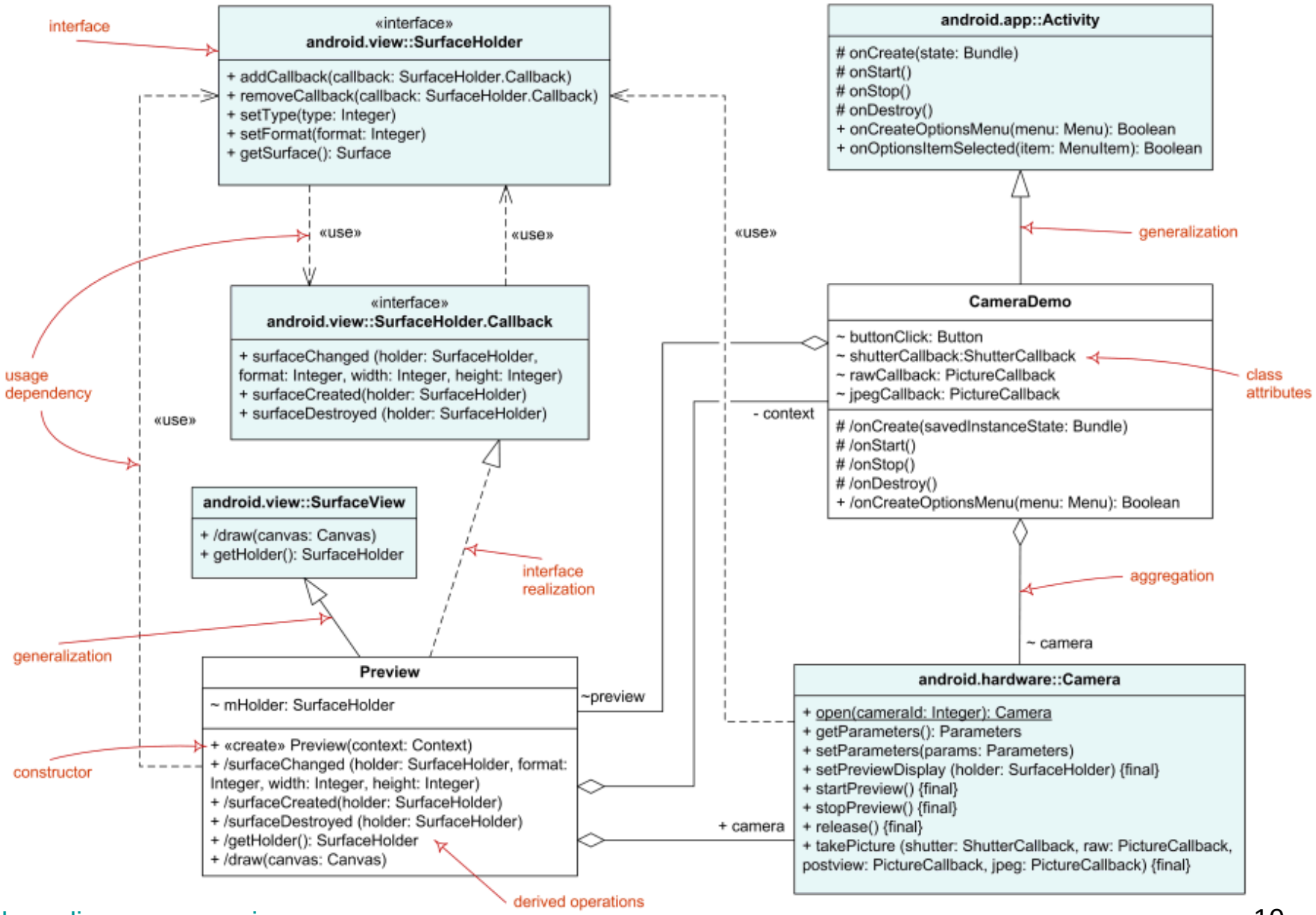
gauss12.js | iframe.html | uml-4.txt | uml-2.txt | task1-1.txt | uml-fdd.txt | new 1 | cat-1 rus.txt

```
1  var diagramColor = 'blueDiagram';
2  var diagramBorder = '';
3  var startTime;
4
5
6  /**
7   * Demonstrate Central limit theorem.
8   */
9  function count() {
10     startTime = new Date().getTime();
11
12     var n = parseInt(document.getElementById("number_of_sources").value) || 0;
13     var k = parseInt(document.getElementById("number_of_elements").value) || 0;
14
15     clearHistogram();
16     clearResult();
17
18     if(n === 0 || k === 0) {
19         alert("Пожалуйста введите данные!");
20         return;
21     }
22
23     var generatedNumbers = generate(n, k);
24
25     var array = generatedNumbers.array;
26     var min = generatedNumbers.min;
27     var expectedValue = generatedNumbers.expectedValue;
28     var deviation = generatedNumbers.deviation;
29     var gradY = generatedNumbers.gradY;
30
31
32     showHistogram(array, min, gradY);
33     showResult(k, array, expectedValue, deviation);
34 }
```

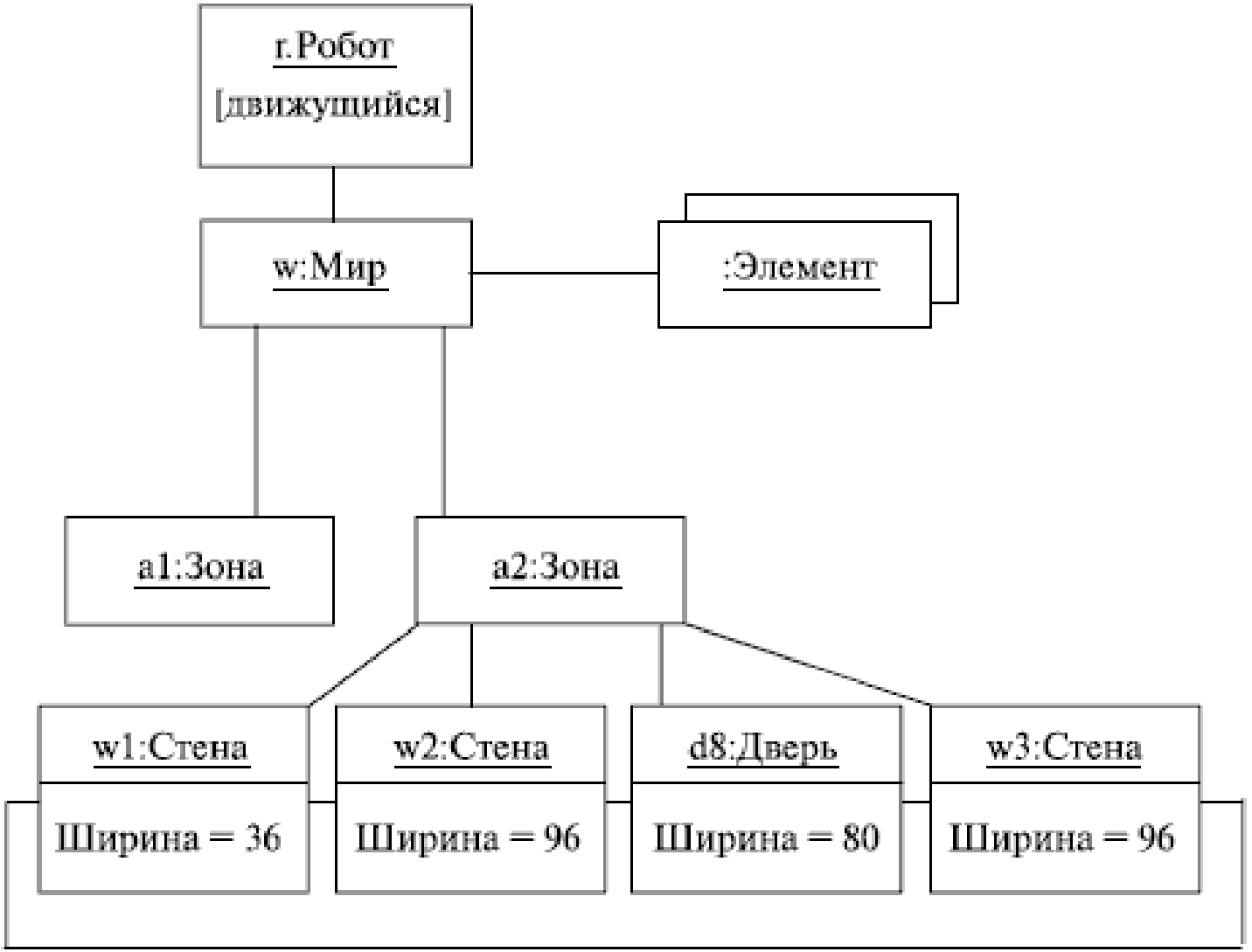
Неявные классы: объявления объектов

```
72      for(var i = 0; i < numberOfElements; i++) {
73          max = normalDistributedNumbers[i] > max ? normalDistributedNumb
74          min = normalDistributedNumbers[i] < min ? normalDistributedNumb
75      }
76      var gradY = (max - min) / 10;
77
78      var countList = [0,0,0,0,0,0,0,0,0,0];
79      for(var k = 0; k < numberOfElements; k++) {
80          var round = Math.floor(normalDistributedNumbers[k] / gradY - (m
81          countList[(round == 10) ? 9 : round] ++; // [1,2)-[2,3)-[3,4)-[
82      }
83
84      return {
85          array           : countList,
86          min             : min,
87          expectedValue   : expectedValue,
88          deviation       : deviation,
89          gradY           : gradY
90      };
91  }
```

Диаграммы классов (Class Diagram)



Диаграммы объектов (Object Diagram)



https://ru.wikipedia.org/wiki/UML
http://www.intuit.ru/studies/courses/1007/229/lecture/5950

Слой в многоуровневой архитектуре

В программной инженерии **многоуровневая архитектура** или **многослойная архитектура** — клиент-серверная архитектура, в которой разделяются функции:

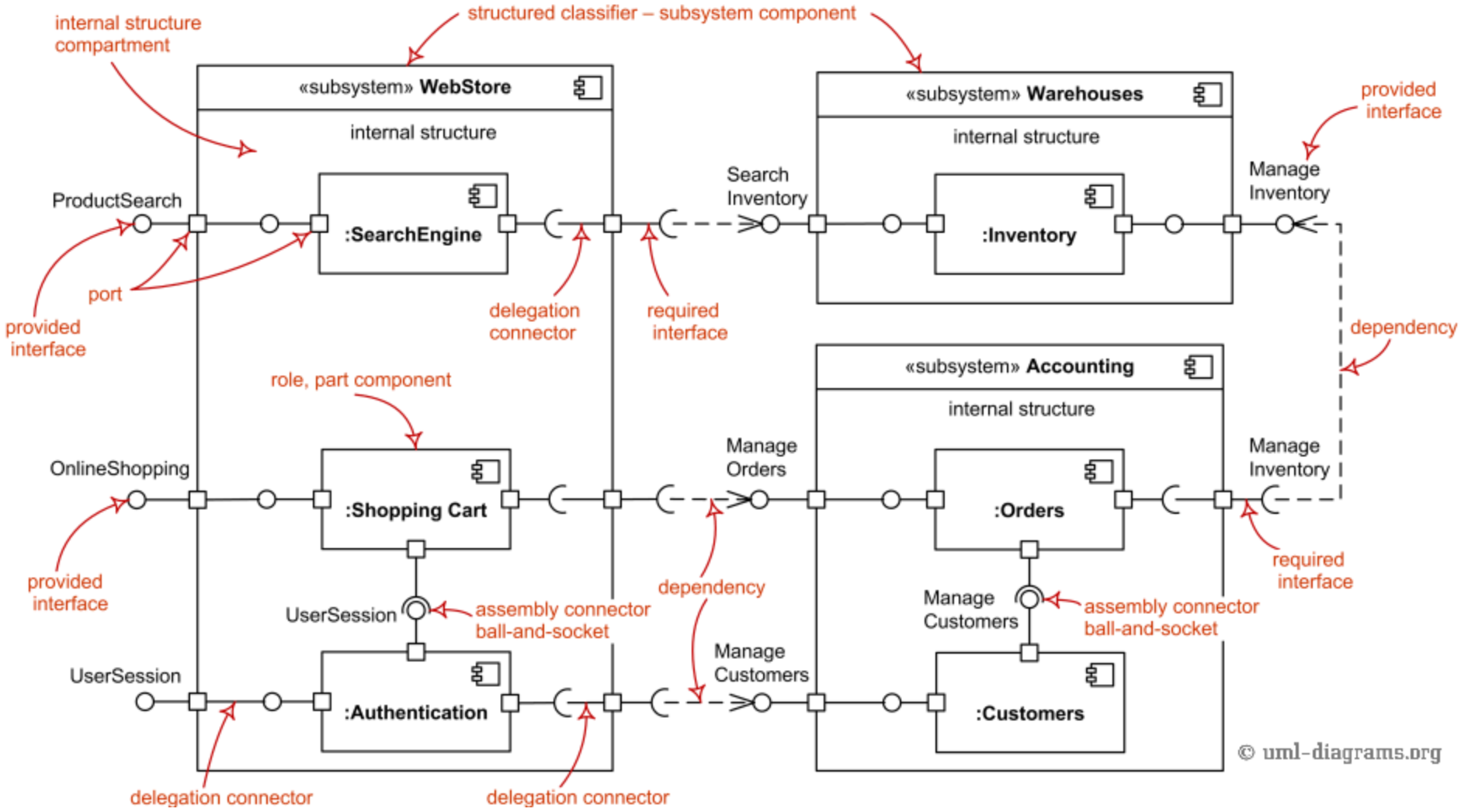
- представления,
- обработки и
- хранения данных.

Разделяя приложение на уровни абстракции, разработчики приобретают возможность внесения изменений в какой-то определённый слой, вместо того, чтобы перерабатывать всё приложение целиком. Архитектурный шаблон «Слои» (англ. *Layers*) помогает структурировать приложения разложением на группы подзадач, находящихся на определенных уровнях абстракции

В логически разделённых на слои архитектурах информационных систем наиболее часто встречаются следующие **четыре слоя:**

- **Слой представления** (слой UI, UIL, пользовательский интерфе́йс, уровень представления в многоуровневой архитектуре)
- **Слой приложения** (сервисный слой, сервисный уровень или GRASP уровень управления)
- **Слой бизнес-логики** (слой предметной области, BLL, доменный слой)
- **Слой доступа к данным** (слой хранения данных, DAL, слой инфраструктуры; логирование, сетевые взаимодействия и другие сервисы, требующиеся для поддержания конкретного слоя бизнес-логики)

Диаграмма компонентов



http://www.uml-diagrams.org/component-diagrams.html

Слои в многоуровневой архитектуре

Трёхуровневая архитектура обычно состоит из:

- уровня ***представления***,
- уровня ***бизнес логики*** и
- уровня ***хранения данных***

Хотя понятия слоя и уровня зачастую используются как взаимозаменяемые, многие сходятся во мнении, что между ними всё-таки есть различие:

слой

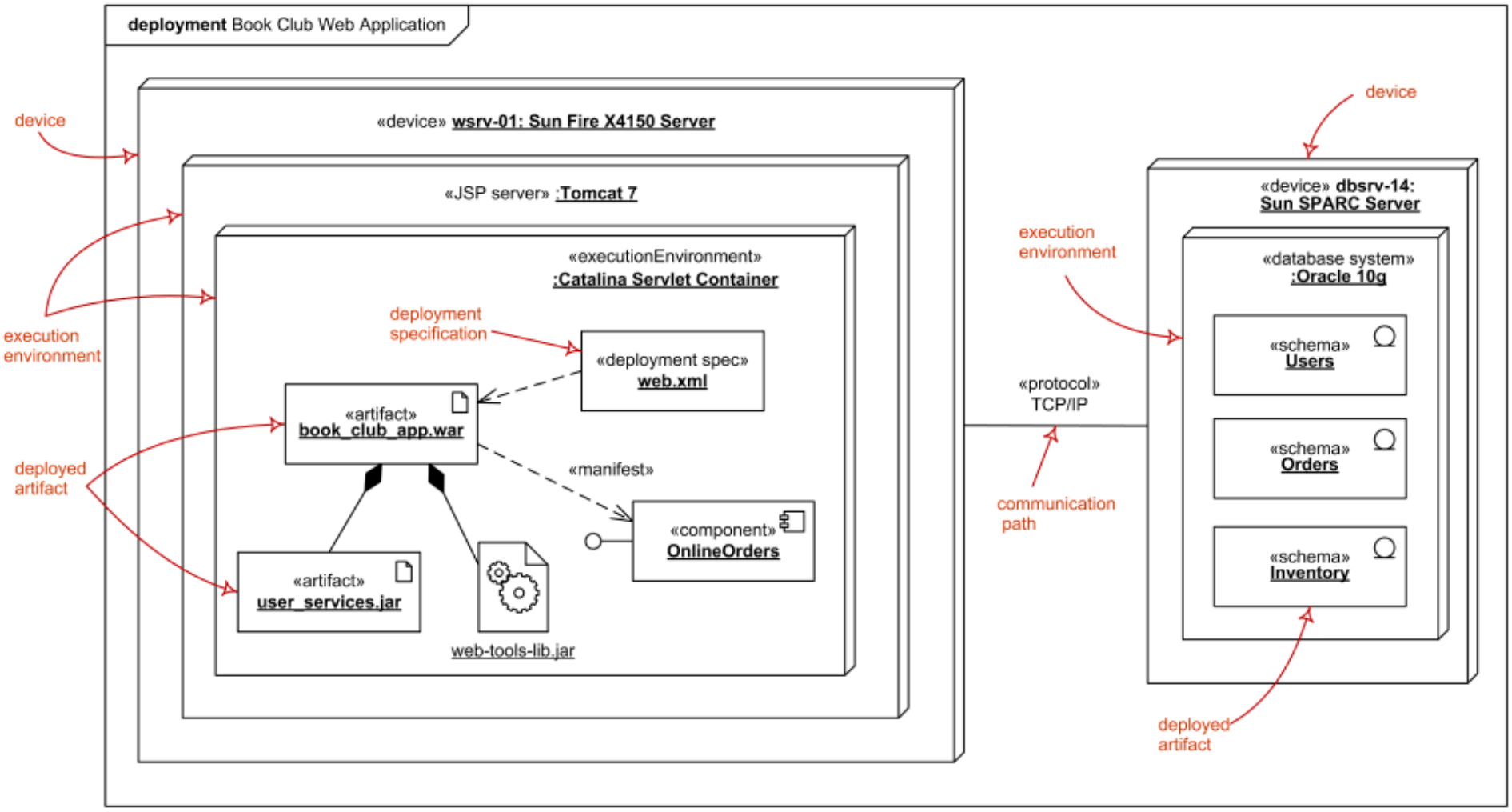
это механизм **логического** структурирования компонентов, из которых состоит программное решение

уровень

это механизм **физического** структурирования инфраструктуры системы

Трёхуровневое решение легко может быть развёрнуто на единственном уровне, таком как персональная рабочая станция

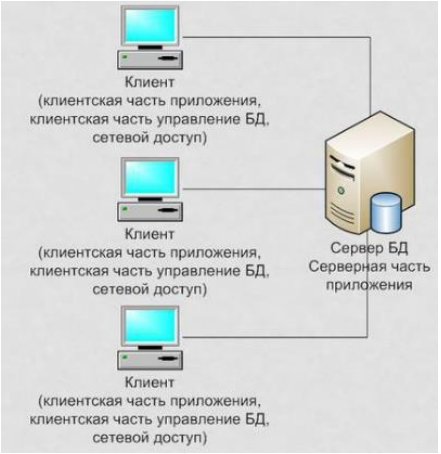
Диаграмма развертывания



http://www.uml-diagrams.org/deployment-diagrams-overview.html

Многослойность

Клиент-сервер (англ. *Client-server*) — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.



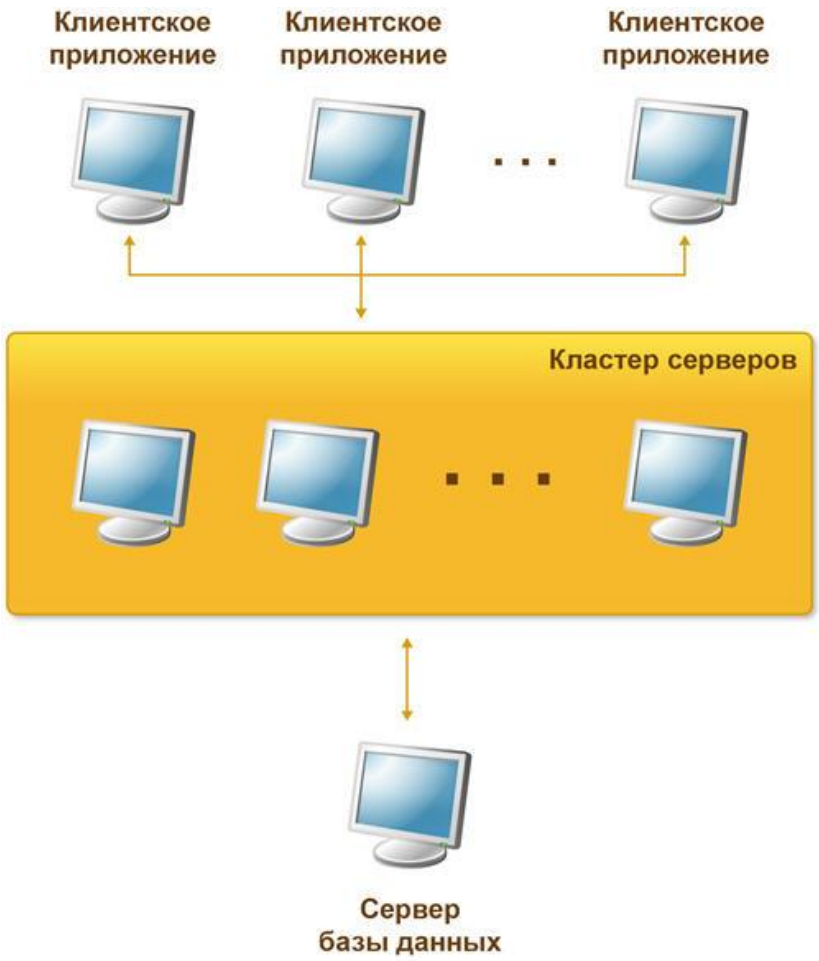
Трёху́ровневая архитекту́ра (*трёхзве́нная архитекту́ра*, англ. *three-layer*) — архитектурная модель программного комплекса, предполагающая наличие в нём трёх компонентов: клиента, сервера приложений (к которому подключено клиентское приложение) и сервера баз данных (с которым работает сервер приложений).баз.

Клиент-сервер
Трехуровневая архитектура

Многослойность

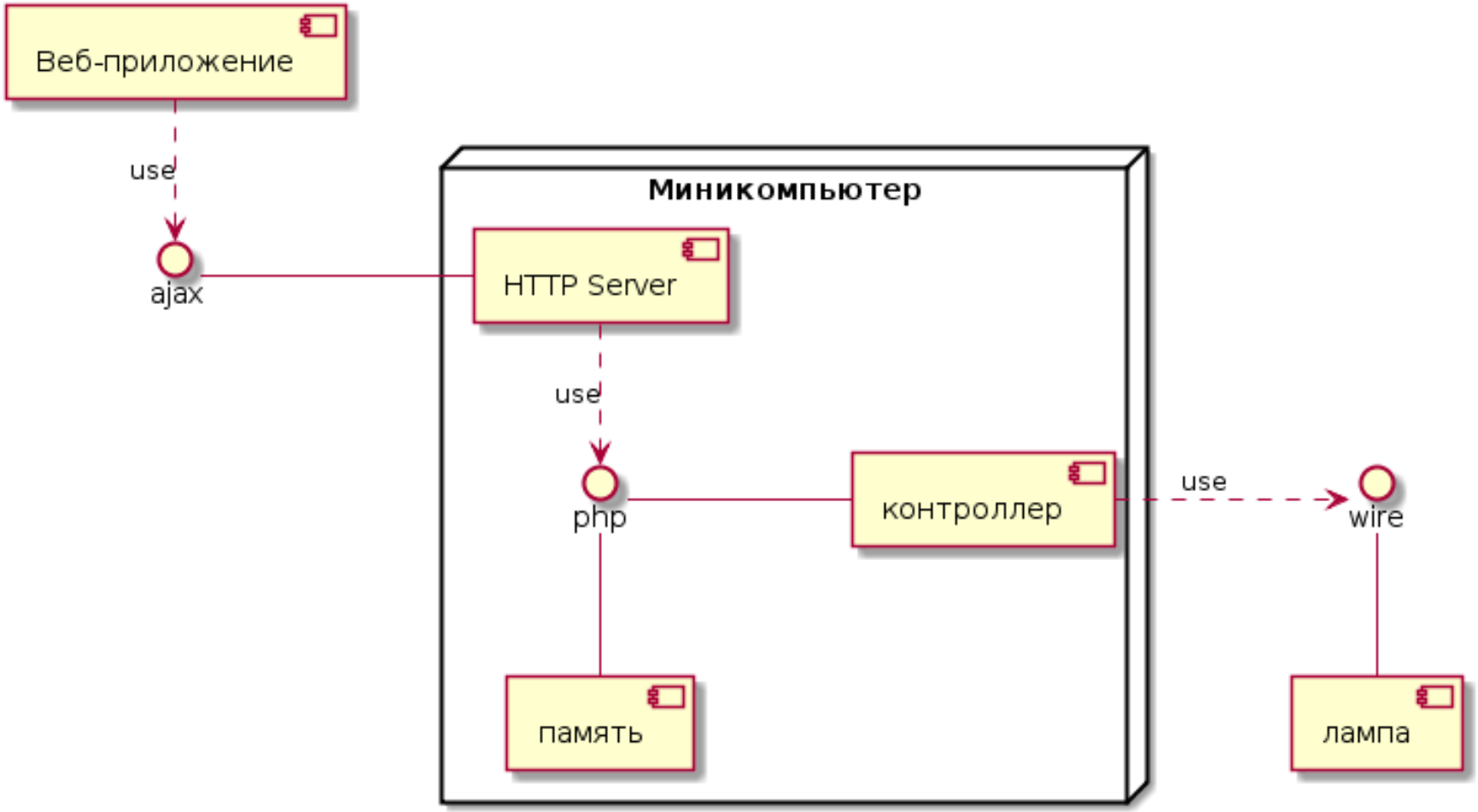
Кластер серверов 1С:Предприятия 8 - основной компонент платформы, обеспечивающий взаимодействие между пользователями и системой управления базами данных в клиент-серверном варианте работы. Наличие кластера позволяет обеспечить бесперебойную, отказоустойчивую, конкурентную работу большого количества пользователей с крупными информационными базами.

Кластер серверов 1С:Предприятия 8 является логическим понятием и представляет собой совокупность рабочих процессов, обслуживающих один и тот же набор информационных баз.

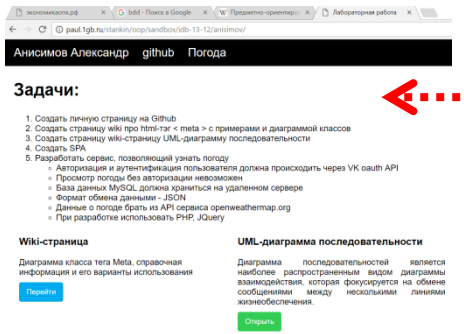


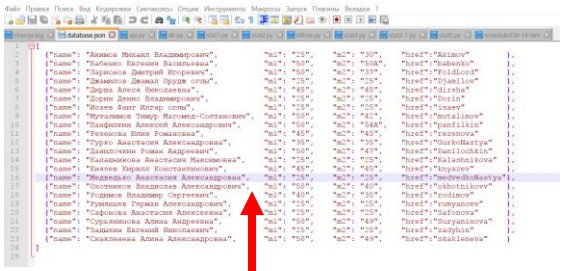
http://v8.1c.ru/overview/Term_000000126.htm

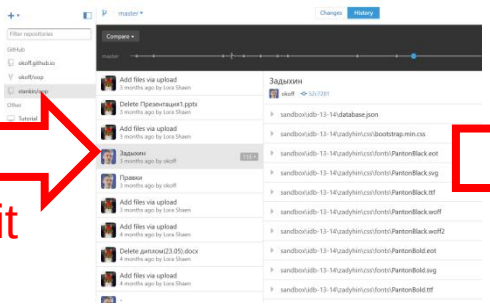
Простое web-приложение

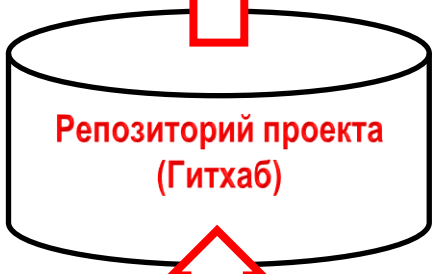


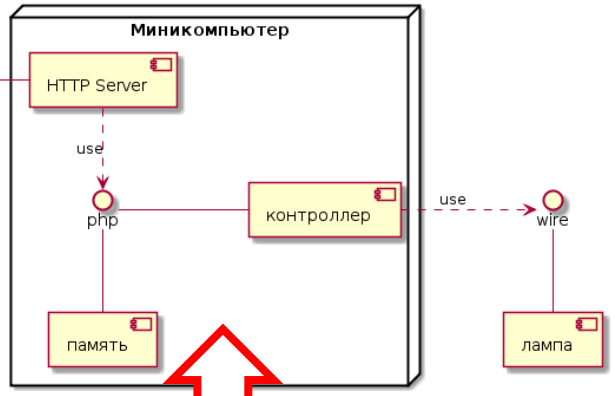
Среда разработки и среда выполнения











Deploy

Commit

Sync

Интернет: браузеры и URL

Бра́узер, или **веб-обозреватель** — прикладное программное обеспечение для просмотра веб-страниц, содержания веб-документов, компьютерных файлов и их каталогов; управления веб-приложениями; а также для решения других задач.

HTML (от англ. *HyperText Markup Language* — «язык гипертекстовой разметки») — стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

Единый указатель ресурса (англ. *Uniform Resource Locator*) — единообразный локатор (определитель местонахождения) ресурса.

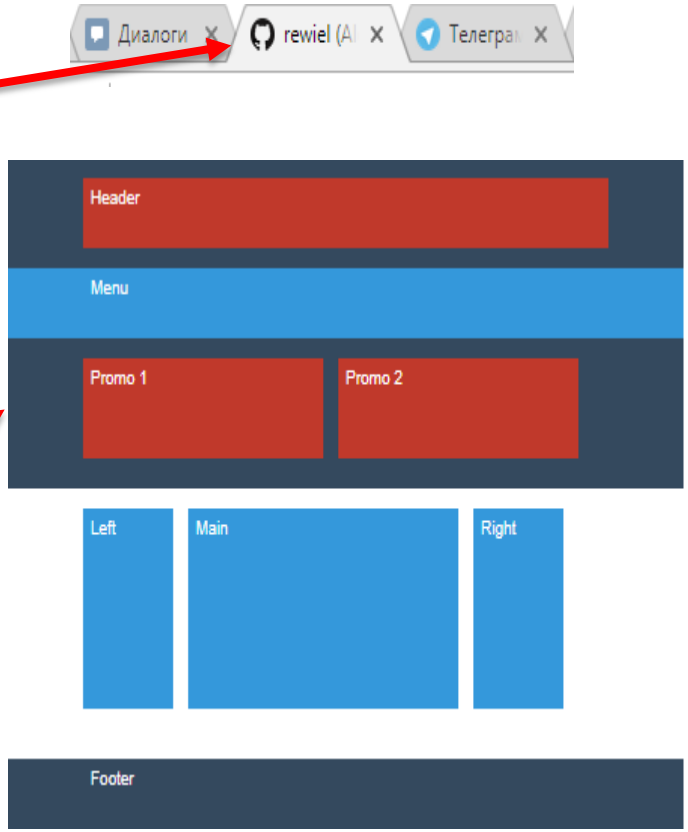
традиционная форма записи URL:
<схема>:[//[<логин>:<пароль>@]<хост>[:<порт>]][/]<URL-путь>[?<параметры>][#<якорь>]

web-верстка

- <head> </head> - голова документа

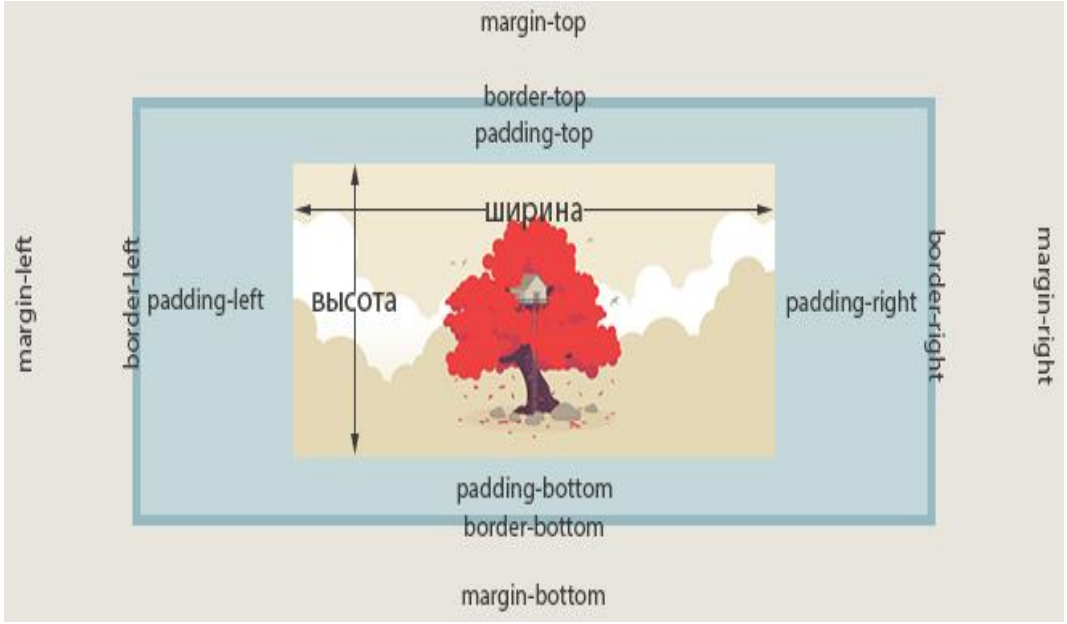
  <title>название</title>

  <body> </body> - тело документа
- Все тэги, расположенные между <head> </head> что-то вроде служебной информации
- Все тэги, расположенные между <body> </body> - непосредственное содержание документа.



Блочные элементы

В блочной модели элемент рассматривается как **прямоугольный контейнер**, имеющий область содержимого, необязательные рамки и отступы (внутренние и внешние).



Блочные элементы — элементы высшего уровня, которые форматируются визуально как блоки, располагаясь на странице в окне браузера вертикально. Основные блочные элементы: **div, p, table, list-item**

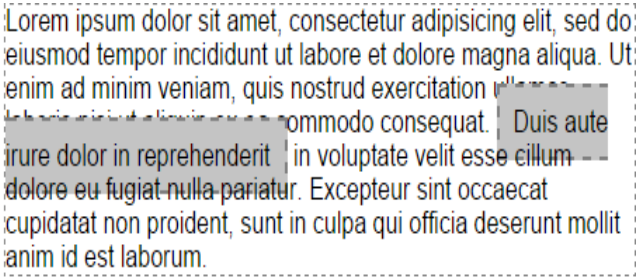
Строчные элементы

Встроенные (строчные) элементы генерируют внутристрочные контейнеры. Они не формируют новые блоки контента.

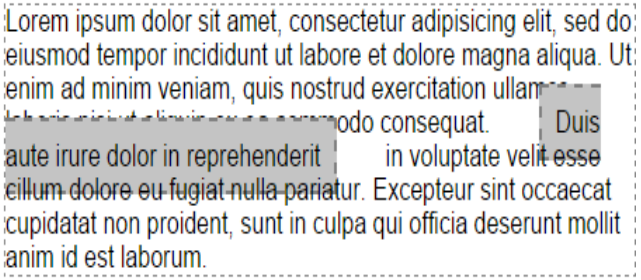
Строчные элементы являются **потомками блочных** элементов.

Ширина и высота строчного элемента зависит только от его содержимого

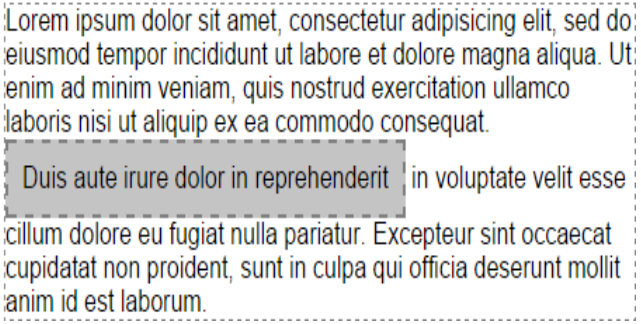
```
span {padding: 10px;
background: #c4c4c4;
border: 2px dashed grey}
```




```
span {padding: 10px;
margin: 30px;
background: #c4c4c4;
border: 2px dashed grey;}
```




```
span {display: inline-block;
padding: 10px;
background: #c4c4c4;
border: 2px dashed grey;}
```



Данные на клиенте: DOM

DOM (от англ. *Document Object Model* — «объектная модель документа») — это не зависящий от платформы и языка программный интерфейс, позволяющий программам и скриптам получить доступ к содержимому HTML-, XHTML- и XML-документов, а также изменять **содержимое**, **структуру** и **оформление** таких документов

Модель DOM не накладывает ограничений на структуру документа

Любой документ известной структуры с помощью DOM может быть представлен в виде **дерева узлов**, каждый узел которого представляет собой:

- **элемент**
- **атрибут**
- текстовый, графический или **любой другой объект**.

Узлы связаны между собой отношениями «родительский-дочерний» (**целое-часть**).

Интернет: интерактивность на клиенте

Dynamic HTML или **DHTML** — это способ (подход) создания **интерактивного** веб-сайта, использующий сочетание

- статичного языка разметки HTML,
- встраиваемого и выполняемого на стороне клиента скриптового языка JavaScript
- CSS (каскадных таблиц стилей) и
- DOM (объектной модели документа)

Он может быть использован для создания приложения в веб-браузере, например, для более простой навигации или для придания интерактивности форм

DHTML может быть использован для динамического перетаскивания элементов по экрану

Также он может служить как инструмент для создания основанных на браузере графических приложений и видеоигр

Веб-разработка: AJAX

AJAX базируется на использовании технологий:

- динамического обращения к серверу «на лету», без перезагрузки всей страницы полностью, например с использованием XMLHttpRequest и
- использования DHTML для динамического изменения содержания страницы

XMLHttpRequest (XMLHTTP, XHR) — API, доступный в скриптовых языках браузеров, таких как JavaScript

Использует запросы HTTP или HTTPS напрямую к веб-серверу и загружает данные ответа сервера напрямую в вызывающий скрипт

Информация может передаваться в любом текстовом формате, например, в XML, HTML или JSON. Позволяет осуществлять HTTP-запросы к серверу без перезагрузки страницы

HTMLHttpRequest (Википедия)

Взаимодействие с http-сервером: AJAX

```
<!DOCTYPE html>
<html>
<head>
<script>
function showHint(str) {
    if (str.length == 0) {
        document.getElementById("txtHint").innerHTML = "";
        return;
    } else {
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function() {
            //alert(this.readyState+' '+this.status);
            if (this.readyState == 4 && this.status == 200) {
                document.getElementById("txtHint").innerHTML = this.responseText;
            }
        }
        xmlhttp.open("GET", "dummy.php?q="+str, true);
        xmlhttp.send();
    }
}
</script>
</head>
<body>

<p><b>Введите данные в строку:</b></p>
<form>
Строка: <input type="text" onkeyup="showHint(this.value)">
</form>
<p>Ответ:<br><span id="txtHint"></span></p>
</body>
</html>
```

Взаимодействие с http-сервером: AJAX

```php
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $parm = $_POST;
    echo('Метод POST ('.count($parm).'):<br>');
} else {
    $parm = $_GET;
    echo('Метод GET ('.count($parm).'):<br>');
}
foreach($parm as $k=>$v) {
    echo('<b>'.$k.'</b> = '.trim(stripslashes(htmlspecialchars($v))).'<br>');
}
?>
```

Взаимодействие с http-сервером: AJAX

jQuery ajax() Method

‹ jQuery AJAX Methods

Example

Change the text of a element using an AJAX request:

```
$("button").click(function(){
  $.ajax({url: "demo_test.txt", success: function(result){
    $("#div1").html(result);
  }});
});
```

Try it Yourself »

JQuery AJAX

Взаимодействие с http-сервером: cURL

cURL — (распространяемая по лицензии MIT), кроссплатформенная служебная программа командной строки, позволяющая взаимодействовать с множеством различных серверов по множеству различных протоколов с синтаксисом URL

Программа cURL может автоматизировать передачу файлов или последовательность таких операций. Например, это хорошее средство для **моделирования действий пользователя** в веб-обозревателе.

Программа поддерживает протоколы:
FTP, FTPS, HTTP, HTTPS, TFTP, SCP, SFTP, Telnet, DICT, LDAP, а также POP3, IMAP и SMTP.

Также cURL поддерживает сертификаты HTTPS, методы HTTP POST, HTTP PUT, загрузку на FTP, загрузку через формы HTTP.

Поддерживаемые методы аутентификации: базовая, дайджест, NTLM и Negotiate для HTTP, а также Kerberos для FTP.

Взаимодействие с http-сервером: WebSocket

WebSocket — протокол связи поверх TCP-соединения, предназначенный для обмена сообщениями между браузером и веб-сервером в режиме реального времени (online)

В настоящее время в W3C осуществляется стандартизация API Web Sockets. Черновой вариант стандарта этого протокола утверждён IETF

WebSocket разработан для воплощения в веб-браузерах и веб-серверах, но он может быть использован для любого клиентского или серверного приложения.

Протокол WebSocket — это независимый протокол, основанный на протоколе TCP. Он делает возможным более тесное взаимодействие между браузером и веб-сайтом, способствуя распространению интерактивного содержимого и созданию приложений реального времени

Распределенные вычисления

Распределённые вычисления — способ решения трудоёмких вычислительных задач с использованием нескольких компьютеров, чаще всего объединённых в параллельную вычислительную систему

Грид-вычисления (англ. *grid* — решётка, сеть) — это форма распределённых вычислений, в которой «виртуальный суперкомпьютер» представлен в виде кластеров, соединённых с помощью сети, слабосвязанных гетерогенных компьютеров, работающих вместе для выполнения огромного количества заданий (операций, работ). Эта технология применяется для решения научных, математических задач, требующих значительных вычислительных ресурсов. Грид-вычисления используются также в коммерческой инфраструктуре для решения таких трудоёмких задач, как экономическое прогнозирование, сейсмоанализ, разработка и изучение свойств новых лекарств.

BOINC (англ. ***Berkeley Open Infrastructure for Network Computing***) — открытая программная платформа (университета Беркли для GRID вычислений) — некоммерческое межплатформенное ПО для организации распределённых вычислений. Используется для организации добровольных вычислений

Распределенные вычисления (Википедия) *Облачные архивы*

Параллельные алгоритмы

Параллельные вычисления — способ организации компьютерных вычислений, при котором программы разрабатываются как набор взаимодействующих вычислительных процессов, работающих параллельно (одновременно)

Термин охватывает совокупность вопросов параллелизма в программировании, а также создание эффективно действующих аппаратных реализаций

Теория параллельных вычислений составляет раздел прикладной теории алгоритмов

Параллельные программы могут физически исполняться либо последовательно на единственном процессоре — перемежая по очереди шаги выполнения каждого вычислительного процесса, либо параллельно — выделяя каждому вычислительному процессу один или несколько процессоров (находящихся рядом или распределённых в компьютерную сеть).

Основная сложность при проектировании параллельных программ — обеспечить правильную последовательность взаимодействий между различными вычислительными процессами, а также **координацию** ресурсов, разделяемых между процессами

Параллельные вычисления (Википедия)

Интернет: веб-сервер

Веб-сервер — сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-потоком или другими данными.

Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и непосредственно компьютер (см.: Сервер (аппаратное обеспечение)), на котором это программное обеспечение работает.

Веб-серверы могут иметь различные дополнительные функции, например:

- автоматизация работы веб-страниц;
- ведение журнала обращений пользователей к ресурсам;
- аутентификация и авторизация пользователей;
- поддержка динамически генерируемых страниц;
- поддержка HTTPS для защищённых соединений с клиентами.

В терминологии компьютерных сетей, **балансировка нагрузки**, или **выравнивание нагрузки** (англ. *load balancing*) — метод распределения заданий между несколькими сетевыми устройствами (например, серверами) с целью оптимизации использования ресурсов, сокращения времени обслуживания запросов

Интернет: сервер приложений

Сервер приложений (англ. *application server*) — это программная платформа (фреймворк), предназначенная для эффективного исполнения процедур (программ, скриптов), на которых построены приложения.

Сервер приложений действует как набор компонентов, доступных разработчику программного обеспечения через API (интерфейс прикладного программирования), определённый самой платформой.

Для веб-приложений основная задача компонентов сервера — обеспечивать создание динамических страниц. Однако современные серверы приложений включают в себя и поддержку кластеризации, повышенную отказоустойчивость, балансировку нагрузки, позволяя таким образом разработчикам сфокусироваться только на реализации бизнес-логики

В случае Java-сервера приложений, сервер приложений ведёт себя как расширенная виртуальная машина для запуска приложений, прозрачно управляя соединениями с базой данных, с одной стороны, и соединениями с веб-клиентом, с другой

Интернет: веб-служба

Веб-служба, *веб-сервис* (англ. *web service*) — идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя

Веб-службы могут взаимодействовать друг с другом и со сторонними приложениями посредством сообщений, основанных на определённых протоколах (SOAP, XML-RPC и т. д.) и соглашениях (REST)

Веб-служба является единицей модульности при использовании сервис-ориентированной архитектуры приложения

В обиходе **веб-сервисами** (**e-services**, electronic services) называют услуги, оказываемые в Интернете. В этом употреблении **термин требует уточнения**, идёт ли речь о поиске, веб-почте, хранении документов, файлов, закладок и т. п. Такими веб-сервисами можно пользоваться независимо от компьютера, браузера или места доступа в Интернет

Интернет: сервис-ориентированная архитектура

ГОСТ Р ИСО/МЭК 18384-1-2017 Информационные технологии (ИТ). Эталонная архитектура для сервис-ориентированной архитектуры (SOA RA). Часть 1. Терминология и концепции SOA

служба (service):

логическое представление ряда **действий**, который имеет определенные результаты, является автономным, может быть составлен из других служб и является "черным ящиком" для потребителей данной службы.

сервис-ориентированная архитектура (service oriented architecture, SOA):

архитектурный стиль, который поддерживает **ориентированность на службы** и является парадигмой для создания бизнес-решений.

Службы, реализованные в этом стиле, используют действия, включающие в себя **бизнес-процессы**, имеют описания для обеспечения контекста, могут быть реализованы через **композицию служб**, могут иметь реализации, зависящие от окружения, описанные в контексте, который их ограничивает либо дает им дополнительные возможности, требуют управления и налагают требования на инфраструктуру для достижения функциональной совместимости и независимости от местоположения с использованием стандартов в максимально возможной степени.

Интернет: микросервисная архитектура

Микросервисная архитектура — **вариант** сервис-ориентированной архитектуры программного обеспечения, направленный на взаимодействие насколько это возможно **небольших**, слабо связанных и легко изменяемых **модулей** — ***микросервисов***, получивший распространение в середине 2010-х годов в связи с развитием практик гибкой разработки и DevOps.

Если в традиционных вариантах сервис-ориентированной архитектуры модули могут быть сами по себе достаточно **сложными** программными системами, а взаимодействие между ними зачастую полагается на стандартизованные тяжеловесные протоколы (такие, как SOAP, XML-RPC), в микросервисной архитектуре системы выстраиваются из компонентов, выполняющих относительно **элементарные функции**, и взаимодействующие с использованием **экономичных** сетевых коммуникационных **протоколов** (в стиле REST с использованием, например, JSON, Protocol Buffers, Thrift).

За счёт повышения гранулярности модулей архитектура нацелена на уменьшение степени зацепления и увеличение связности, что позволяет проще добавлять и изменять функции в системе в любое время.

Интернет: службы облачных вычислений

ГОСТ ISO/IEC 17788-2016 Информационные технологии (ИТ). Облачные вычисления. Общие положения и терминология

облачные вычисления (cloud computing):
парадигма для предоставления возможности **сетевого доступа** к масштабируемому и эластичному пулу общих физических или виртуальных ресурсов с предоставлением самообслуживания и администрированием по требованию

соглашение об уровне обслуживания (service level agreement, SLA):
письменное соглашение между поставщиком и заказчиком, в котором задокументированы **услуги** и согласованы **уровни услуг**

Таблица А.1 — Категории служб облачных вычислений и типы возможностей облака

Категория служб облачных вычислений	Тип возможностей облака		
	Инфраструктура	Платформа	Приложение
Вычисления как услуга	X		
Обмен информацией как услуга		X	X
Хранение данных как услуга	X	X	X
Инфраструктура как услуга	X		
Сеть как услуге	X	X	X
Платформа как услуга		X	
Программное обеспечение как услуга			X

Интернет: службы облачных вычислений

служба облачных вычислений (cloud service):
одна или более возможностей, предоставляемых через **облачные вычисления**, вызываемая посредством определенного интерфейса

программное обеспечение как услуга (SaaS) [software as a service (SaaS)]:
категория служб облачных вычислений, в которой потребитель службы облачных вычислений может использовать **приложения** поставщика службы облачных вычислений