# Введение в базовые понятия IT

В предыдущих темах мы разобрали фундаментальные стороны IT-проекта: жизненный цикл проекта, подходы к его разработке, роли и функции разных специалистов. Теперь стоит остановиться и более внимательно изучить техническую сторону IT-проекта, так как ни один участник команды разработки не выполняет свои задачи в изоляции от других. Между специалистами происходит постоянный обмен знаниями, благодаря которому в команде появляется общий контекст и нарабатывается опыт.

Знания и термины из этого раздела универсальны: так или иначе, вы будете сталкиваться с ними во время работы над проектами, и их понимание поможет вам лучше справляться со своими задачами. Желаем успехов!

Главные термины раздела, о которых будет идти речь ниже:

:info:**DNS-сервер** — компьютер, который хранит или кэширует IP-адреса сайтов и выдаёт их браузеру по запросу.

:info:**Провайдер** — компания, которая обеспечивает доступ к интернету.

:info:**IP-адрес** — цифровой идентификатор, присваиваемый серверу.

:info:**Роутер** — маршрутизатор. Устройство, которое обеспечивает подключение компьютеров к интернету.

:info:**Домен** — уникальное имя сайта.

:info:**Hosting-провайдеры** — организации, предоставляющие сервера в аренду, чтобы вы могли разместить сайт в сети.

:info:**Cloud или облако, облачные серверы** — сеть мощных компьютеров — серверов, которые позволяют клиентам пользоваться своими ресурсами через интернет.

# Общая схема информационных технологий

:info:**Информационные технологии** — процессы, использующие совокупность средств и методов сбора, обработки, накопления и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса, явления, информационного продукта, а также распространение информации и способы осуществления таких процессов и методов (согласно Федеральному Закону «Об информации, информационных технологиях и о защите информации» № 149-ФЗ).



Или, более кратко: процесс, использующий совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления (информационного продукта).

Когда термин **«технология»** стал применяться к информационным процессам, возникло понятие «информационная технология» — совокупность знаний о способах автоматизированной переработки информации с использованием ЭВМ для автоматизации управленческой деятельности.

Информационные технологии (ИТ) четко делятся на два компонента:

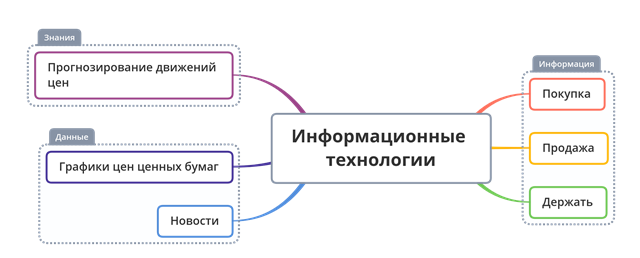
* Аппаратное обеспечение: физическая структура, конфигурация вычислительной техники, систем и прочего оборудования.
* Программное обеспечение: набор правил, руководящих принципов и алгоритмов, необходимых для обеспечения работоспособности технического оборудования.

:info:**Данные** — факты реальной жизни.

:info:**Информация** — обработанные данные, используемые для принятия решений и решения задач, сведения, независимо от формы их представления.

:info:**Знания** — обработанная информация, используемая для принятия решений, решения задач и создания новой информации.

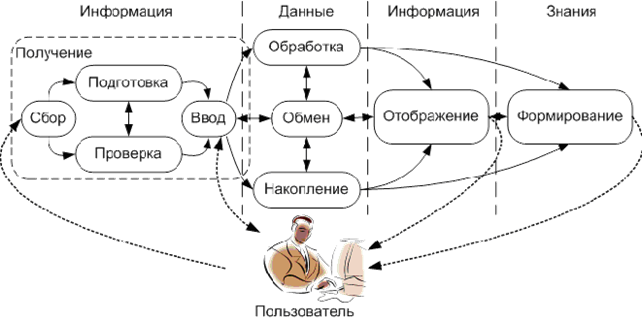
Например, на фондовом рынке:



В качестве **данных** выступают значения цен бумаг в определенный момент времени. Например, 8 октября 2021 г. обыкновенные акции Microsoft стоили $294,74 за акцию. Кроме того, к данным относятся новости, влияющие на рынок ценных бумаг. Например, новость о разливе нефти в Мексиканском заливе влияет на цену акций компании British Petroleum.

**Информация** на фондовом рынке — сопоставление альтернатив, имеющихся у лица, торгующего ценными бумагами. Человек может купить ценную бумагу, продать ее или держать ее в своем портфеле. Основа для сопоставления альтернатив — ожидаемое движение цены.

Наконец, **знания** — всё то, что позволяет участнику рынка более эффективно преобразовывать данные в информацию.



В современном обществе основным технологическим средством накопления, переработки, защиты и обмена информации служат **компьютеры и средства телекоммуникации.**

Внедрение компьютеров в информационную сферу и распространение телекоммуникационных средств связи в свое время определили новый этап развития ИТ.

Таким образом, **информационная технология** — емкое понятие, которое отражает представления информационного общества о процессах преобразования и потребления информации.

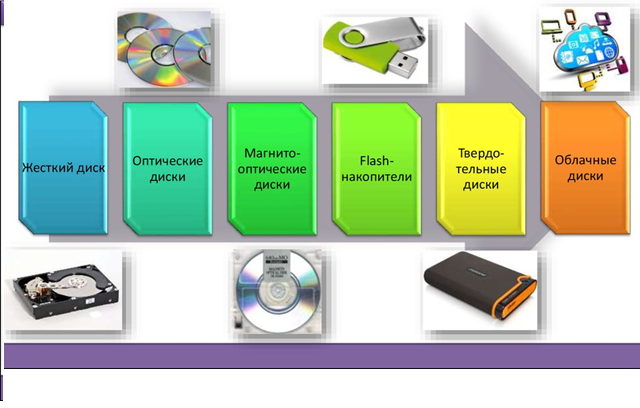
# Информация: хранение и обработка



Основные операции, производимые над информацией — хранение и обработка.

**Хранение**

Рост потока информации вследствие прогресса совпадает с развитием технических средств хранения данных:



:info:**Хранение информации** — обеспечение сохранности документов или данных, поддержание неизменно удовлетворительного состояния материального носителя, защита от несанкционированного доступа и недозволенного использования.

Первоначально вся информация хранилась только в памяти человека. Первые общественные хранилища информации — библиотеки и архивы — возникли с появлением первых форм документной информации — табличек, рукописей, книг.

Сегодня сохранность документов обеспечивают книжные палаты, видеотеки, фильмотеки, фонотеки, патентные бюро, музеи, картинные галереи, медиатеки, депозитарии. Основные современные хранилища информации компьютерные: базы и банки данных, базы знаний, автоматизированные информационно-поисковые системы, электронные библиотеки.

Чтобы хранить информацию, система должна ее воспринять, переработать и преобразовать в физическое явление, т. е. внести на соответствующий носитель. Хранение информации в оцифрованном виде дает возможность:

* в любой момент обратиться к памяти ЭВМ за системными и другими необходимыми пользователям программами и данными;
* осуществлять различных видов работы на компьютере;
* обеспечивать высокий уровень безопасности данных.

Для долговременного хранения информации важным является выбор соответствующего носителя. С первой половины прошлого века надежными носителями информации считались фотоматериалы — в специальных условиях они могут долго ее сохранять. В процессе эволюции компьютерных технических средств информация хранилась на машинных носителях: перфокартах, перфолентах, магнитных лентах, магнитных дисках и дискетах.

Прогресс подарил нам компактные оптические диски (CD, DVD и др.) и твердотельную флеш-память (более известную как «флешки»). Для осуществления операций записи и хранения на всех этих видах электронных носителей данных используются соответствующие устройства и технологии.

Данные сохраняют, чтобы иметь возможность использовать их в будущем. Для этого создаются локальные, распределенные и удаленные базы данных, информационные хранилища (репозитории) или хранилища данных.

:info:Хранилища данных — предметно ориентированные, интегрированные, неизменчивые, поддерживающие хронологию наборы данных, организованные для целей поддержки управления.

Характерной особенностью сетевого хранилища данных является то, что одновременно к нему с одним и тем же запросом могут обратиться несколько пользователей, которые получат одинаковые сведения.

**Базы данных**

:info:**База данных, БД** — это совокупность структурированных данных определенного назначения.

:info:**Структурирование данных** — это объединение данных по определенным параметрам.

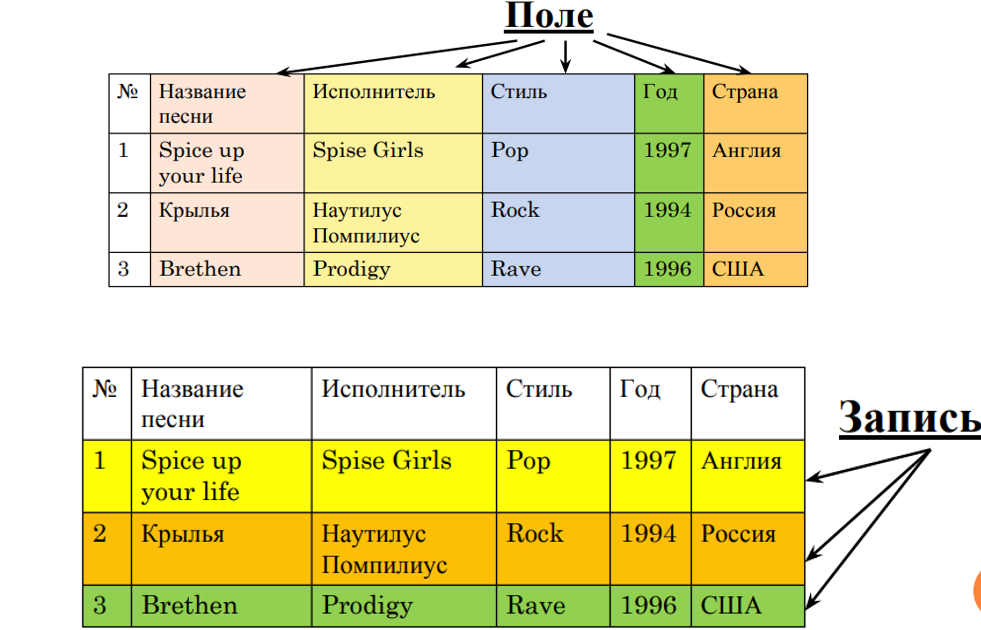
:info:**Реляционная база данных** — база данных, основанная на реляционной модели данных. Понятие «реляционный» основано на англ. relation («отношение, зависимость, связь»).

Существует два основных типа баз данных:

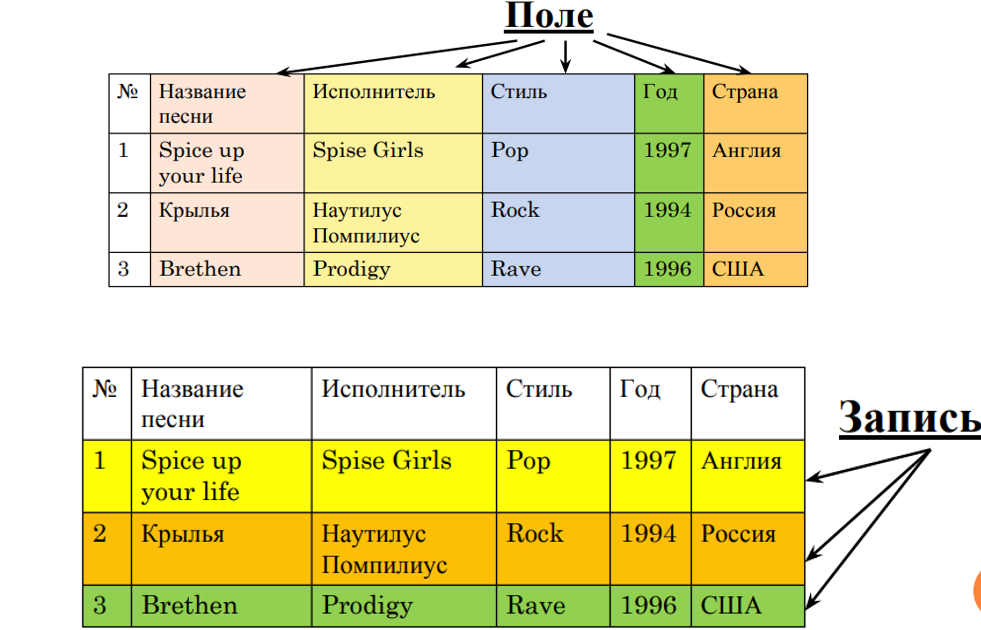
1. **Объектно-ориентированная база данных (ООБД)** — база данных, в которой данные моделируются в виде объектов, их атрибутов, методов и классов.
2. **Объектно-реляционная** **система управления базами данных (ОРСУБД)** — реляционная СУБД (РСУБД). Она поддерживает некоторые технологии, присущие объектно-ориентированным СУБД и реализующие объектно-ориентированный подход: объекты, классы и наследование реализованы в структуре баз данных и языке запросов. Объектно-реляционными СУБД являются, например Oracle Database, Informix, DB2, PostgreSQL.

Данные обязательно должны быть структурированы и связаны между собой так, чтобы человек мог составить представление о каком-либо объекте, явлении или процессе. Основными понятиями БД являются **поле** и **запись**.

:info:**Поле базы данных** — это столбец таблицы, содержащий значения определенного свойства. Поле характеризуется именем и типом данных.



:info:**Запись** — это совокупность значений параметров конкретного объекта.

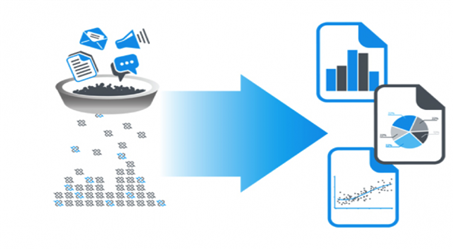


:info:Структура записи — это совокупность логически связанных полей, характеризующих параметры реального объекта.

Общепринятым стандартом языка работы с реляционными базами данных в настоящее время является язык структурированных запросов (Structured Query Language — SQL). Это универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных. Вопреки существующим заблуждениям, SQL — это информационно-логический язык, а не язык программирования.

**Обработка**

Целью любого веб-приложения является обработка данных и предоставление нужной человеку информации.



Обработка данных представляют собой систематизированную последовательность действий, которые начинаются с момента получения информации и проводятся до тех пор, пока не будут получены ожидаемые результаты.

Технологии обработки данных предназначены для решения хорошо структурированных задач, в которых имеются необходимые входные данные и известны алгоритмы и другие стандартные процедуры их обработки.

:info:Обработка данных — создание из поступающих данных информации, отражающей деятельность организации.

В процессе обработки данных используются следующие типовые операции:

* классификация, или группировка;
* сортировка, то есть упорядочивание последовательности записей;
* вычисления, включающие арифметические и логические операции — это дает возможность получать новые данные;
* укрупнение или агрегирование, служащее для уменьшения количества данных и реализуемое в форме расчетов итоговых или средних значений.

Многие технологии подразделяются на разные группы в зависимости от своего предназначения, например:

* обработка текстовой информации;
* обработка графики, растровой и векторной;
* обработка анимации, видео, звука;
* обработка знаний, особые экспертные системы;
* обработка речи и т.п.

# Вычислительные устройства и их взаимодействие

Все существующие современные компьютеры можно разделить на несколько категорий. Каждой из них соответствует специфическая программная инфраструктура:

|  |  |
| --- | --- |
| Карманные персональные компьютеры (КПК) | Портативные вычислительные устройства, которые обладают широкими функциональными возможностями — например, смартфоны. Сейчас уже могут представлять собой полноценный мобильный офис. |
| IoT устройства (internet of things) | Относительно новый класс устройств, физические объекты («вещи»), оснащённые встроенными средствами и технологиями для взаимодействия друг с другом или с внешней средой. Предполагается, что организация сетей из «умных вещей» способна перестроить экономические и общественные процессы, исключить из части действий и операций необходимость участия человека. |
| Портативные компьютеры | Виды компьютеров, которые можно свободно перемещать в любое место. К мобильным компьютерам относятся ноутбуки, нетбуки, планшеты, КПК, iPad. |
| Настольные компьютеры | Самая большая категория представленных на рынке ПК. Активно применяются как корпоративными пользователями (Office PC), так и в сфере домашнего использования (Home PC). |
| Тонкие клиенты | Компьютер в сетях с клиент-серверной или терминальной архитектурой, который переносит задачи по обработке информации на сервер. Не имеет жёсткого диска, использует специализированную локальную ОС, выполняется в специализированных корпусах с полностью пассивным охлаждением. Примером тонкого клиента может служить компьютер с браузером, использующийся для работы с веб-приложениями. |
| Серверы | Специализированные высокопроизводительные компьютеры, способные обслуживать несколько одновременно подключающихся к ним компьютеров для выполнения определенных задач. |
| Суперкомпьютеры | Применяются для решения задач, с которыми не справляются обычные компьютеры. К таким относятся задачи аэродинамики, сейсмологии, атомной и ядерной физики, космических, военных и оборонных исследований, математического моделирования. |
| Кластерная система (кластер) | Группа компьютеров с высокоскоростными каналами связи, которые составляют единый ресурс. Такие системы обеспечивают высокую степень отказоустойчивости за счет возможности мгновенного автоматического перехода с вышедшего из строя узла на работающий. |



Вычислительные сети, необходимые для взаимодействия устройств, подразделяются на разные группы в зависимости от назначения и характеристик:

1. По территориальной распространенности:

* [BAN](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) (Body Area Network);
* [PAN](https://ru.wikipedia.org/wiki/Personal_area_network) (Personal Area Network);
* [LAN](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) (ЛВС, Local Area Network);
* [CAN](https://ru.wikipedia.org/wiki/Campus_Area_Network) (Campus Area Network);
* [MAN](https://ru.wikipedia.org/wiki/%D0%93%D0%BE%D1%80%D0%BE%D0%B4%D1%81%D0%BA%D0%B0%D1%8F_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) (Metropolitan Area Network);
* [WAN](https://ru.wikipedia.org/wiki/%D0%93%D0%BB%D0%BE%D0%B1%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) (Wide Area Network).

1. По архитектуре:

* клиент-сервер;
* одноранговая сеть (децентрализованная или пиринговая сеть).

1. По типу сетевой топологии:

* шина;
* кольцо;
* звезда;
* дерево;
* смешанная топология.

1. По типу среды передачи:

* проводные;
* беспроводные.

1. По функциональному назначению:

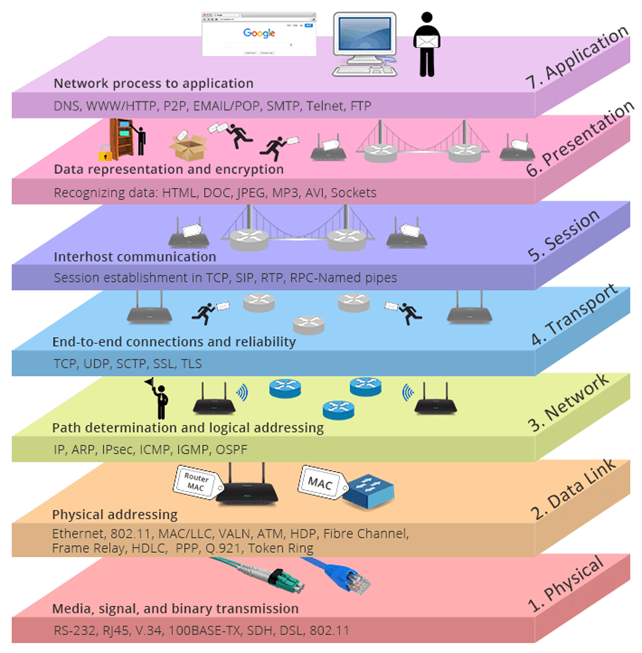
* сети хранения данных;
* серверные фермы;
* сети управления процессом;
* сети SOHO/домовые сети.

1. По скорости передачи:

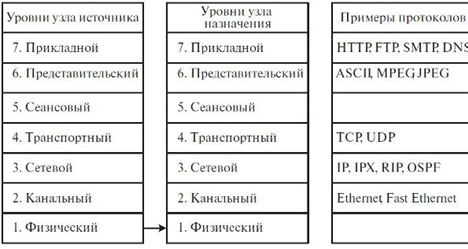
* низкоскоростные (до 10 Мбит/с);
* среднескоростные (до 100 Мбит/с);
* высокоскоростные (свыше 100 Мбит/с).

# Сети, каналы связи, протоколы, пространство имен

В компьютерных сетях идеологической основой стандартизации является **многоуровневый подход** к разработке средств сетевого взаимодействия. Международная организация стандартов (International Standard Organization — ISO) предложила семиуровневую модель открытого взаимодействия систем (Open System INTerconnection), более известную в своем сокращенном варианте — [OSI/ISO](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI).



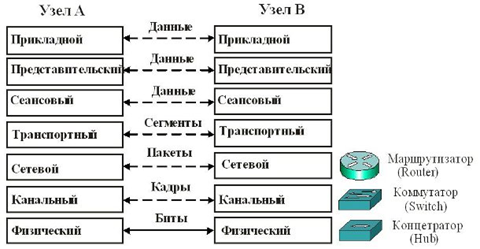
Модель ISO/OSI включает семь уровней. На рисунке ниже показана модель взаимодействия двух устройств: узла источника (source) и узла назначения (destination):



Совокупность правил, по которым происходит обмен данными между программно-аппаратными средствами, находящимися на одном уровне, называется **протоколом.**

Набор протоколов называется **стеком протоколов** и задается определенным стандартом. Взаимодействие между уровнями определяется стандартными интерфейсами.

Взаимодействие соответствующих уровней является виртуальным. Исключение — физический уровень, то есть когда обмен данными происходит по кабелям, соединяющим компьютеры. На рисунке приведены примеры протоколов, управляющих взаимодействием узлов на различных уровнях модели OSI.



Взаимодействие уровней между собой внутри узла происходит через межуровневый интерфейс, и каждый нижележащий уровень предоставляет услуги вышележащему.

Виртуальный обмен между соответствующими уровнями узлов A и B происходит определенными единицами информации. На трех верхних уровнях — сообщения или данные (Data), на транспортном уровне — сегменты (Segment), на сетевом уровне — пакеты (Packet), на канальном уровне — кадры (Frame), а на физическом — последовательность битов.

Для каждой сетевой технологии существуют свои протоколы и свои технические средства, часть из которых имеет условные обозначения, которые были введены фирмой Cisco и стали общепринятыми.

Среди технических средств физического уровня:

* кабели;
* разъемы;
* повторители сигналов (repeater);
* многопортовые повторители или концентраторы (hub);
* преобразователи среды (transceiver) — например, преобразователи электрических сигналов в оптические и наоборот.

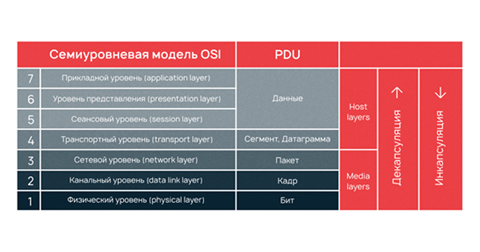
На канальном уровне:

* мосты (bridge);
* коммутаторы (switch).

Сетевые карты или адаптеры (Network Interface Card — NIC) функционируют как на канальном, так и на физическом уровне, что обусловлено сетевой технологией и средой передачи данных.

На сетевом уровне: маршрутизаторы (router).

**Уровни модели ISO/OSI**



1. **Первый, физический уровень (physical layer)**

Самый низкий уровень, который отвечает за обмен физическими сигналами между физическими устройствами. Компьютерное ПО не понимает, что такое картинка или что на ней изображено. Картинка известная ему только в виде набора нулей и единиц, т. е. в битах. В данном случае бит является блоком данных протокола — сокращенно PDU (Protocol Data Unit).

Каждый уровень имеет свои PDU, представляемые в той форме, которая будет понятна на данном уровне и, возможно, на следующем — и так до преобразования. Работа с чистыми данными происходит только на уровнях с пятого по седьмой.

Устройства физического уровня оперируют битами. Они передаются по проводам (например, через оптоволокно) или без проводов (например, через Bluetooth или IRDA, Wi-Fi, GSM, 4G).

1. **Второй уровень, канальный (data link layer)**

Второй уровень решает проблему адресации при передаче информации. Канальный уровень получает биты и превращает их в кадры (frame, «фреймы»). Задача здесь — сформировать кадры с адресом отправителя и получателя, после чего отправить их по сети.

У канального уровня есть два подуровня — это MAC и LLC. MAC (Media Access Control, контроль доступа к среде) отвечает за присвоение физических MAC-адресов, а LLC (Logical Link Control, контроль логической связи) занимается проверкой и исправлением данных, управляет их передачей.

На втором уровне OSI работают коммутаторы, их задача — передать сформированные кадры от одного устройства к другому, используя в качестве адресов только физические MAC-адреса.

1. **Третий уровень, сетевой (network layer)**

На третьем уровне появляется новое понятие — маршрутизация, выполняемая специальными устройствами — маршрутизаторами. Их иногда еще называют роутерами. Маршрутизаторы получают MAC-адрес от коммутаторов с предыдущего уровня и выстраивают маршрут от одного устройства к другому с учетом всех потенциальных неполадок в сети.

На сетевом уровне активно используется протокол ARP (Address Resolution Protocol — протокол определения адреса). С помощью него 64-битные MAC-адреса преобразуются в 32-битные IP-адреса и наоборот, тем самым обеспечивая инкапсуляцию и декапсуляцию данных.

1. **Четвертый уровень, транспортный (transport layer)**

Все семь уровней модели OSI можно условно разделить на две группы:

* Media layers (уровни среды);
* Host layers (уровни хоста).

Уровни группы Media Layers занимаются передачей информации (по кабелю или беспроводной сети) и используются сетевыми устройствами, такими как коммутаторы, маршрутизаторы и т.п. Уровни группы Host layers используются непосредственно на устройствах, будь то стационарные компьютеры или портативные мобильные устройства.

Четвертый уровень — посредник между Host Layers и Media Layers, относящийся при этом скорее к первым, чем к последним. Его главной задачей является транспортировка пакетов. Естественно, при транспортировке возможны потери, но некоторые типы данных более чувствительны к потерям, чем другие. Поэтому при передаче данных, наиболее чувствительных к потерям, на транспортном уровне используется протокол TCP, контролирующий целостность доставленной информации.

Для мультимедийных файлов небольшие потери не так важны, гораздо критичнее будет задержка. Для передачи данных, чувствительных именно к задержкам, используется протокол UDP, позволяющий организовать связь без установки соединения.

При передаче по протоколу TCP данные делятся на сегменты. **Сегмент** — это часть пакета. Когда приходит пакет данных, который превышает пропускную способность сети, пакет делится на сегменты допустимого размера. Сегментация пакетов также требуется в ненадежных сетях, когда существует большая вероятность того, что большой пакет будет потерян или отправлен не тому адресату. При передаче данных по протоколу UDP пакеты данных делятся уже на датаграммы. **Датаграмма** (datagram) — это тоже часть пакета, но ее нельзя путать с сегментом.

Главное отличие датаграмм в автономности. Каждая датаграмма содержит все необходимые заголовки, чтобы дойти до конечного адресата, поэтому они не зависят от сети и могут доставляться разными маршрутами и в разном порядке. Датаграмма и сегмент — это два PDU транспортного уровня модели OSI. При потере датаграмм или сегментов получаются «битые» куски данных, которые не получится корректно обработать.

Первые четыре уровня — специализация сетевых инженеров. Разработчики занимаются пятым, шестым и седьмым.

1. **Пятый уровень, сеансовый (session layer)**

Пятый уровень оперирует чистыми данными; помимо пятого, чистые данные используются также на шестом и седьмом уровне. Сеансовый уровень отвечает за поддержку сеанса или сессии связи. Пятый уровень оказывает услугу следующему: управляет взаимодействием между приложениями, открывает возможности синхронизации задач, завершения сеанса, обмена информации.

Службы сеансового уровня зачастую применяются в средах приложений, требующих удаленного вызова процедур, т. е. чтобы запрашивать выполнение действий на удаленных компьютерах или независимых системах на одном устройстве (при наличии нескольких ОС).

Примером работы пятого уровня может служить видеозвонок по сети: во время видеосвязи необходимо, чтобы два потока данных (аудио и видео) шли синхронно.

1. **Шестой уровень, представления данных (presentation layer)**

Шестой уровень занимается тем, что представляет данные (которые все еще являются PDU) в понятном человеку и машине виде. Например, когда одно устройство умеет отображать текст только в кодировке ASCII, а другое только в UTF-8, перевод текста из одной кодировки в другую происходит именно на шестом уровне.

Шестой уровень также занимается представлением изображений (в JPEG, GIF и т.д.), а также видео и аудио (в MPEG, QuickTime). Помимо перечисленного, шестой уровень занимается шифрованием данных в случаях, когда при передаче их необходимо защитить.

1. **Седьмой уровень, прикладной (application layer)**

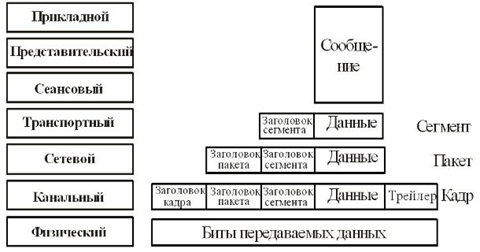
Седьмой уровень иногда еще называют уровнем приложений. Однако чтобы не запутаться, стоит использовать оригинальное название — **application layer.**

Прикладной уровень — это то, с чем взаимодействуют пользователи, своего рода графический интерфейс всей модели OSI.

Все услуги, получаемые седьмым уровнем от других, используются для доставки данных до пользователя. Протоколам седьмого уровня не требуется обеспечивать маршрутизацию или гарантировать доставку данных, когда об этом уже позаботились предыдущие шесть.

Протоколы здесь используют UDP (например, DHCP) или TCP (например, HTTP, HTTPS, SFTP (Simple FTP), DNS). Прикладной уровень является самым верхним по иерархии, но при этом его легче всего объяснить.

В процессе передачи данных важным процессом при передаче данных является **инкапсуляция данных** (encapsulation). Передаваемое сообщение, сформированное приложением, проходит три верхних сетенезависимых уровня и поступает на транспортный уровень, где делится на части и каждая часть инкапсулируется (помещается) в сегмент данных. В заголовке сегмента содержится номер протокола прикладного уровня, с помощью которого подготовлено сообщение, и номер протокола, который будет обрабатывать данный сегмент.



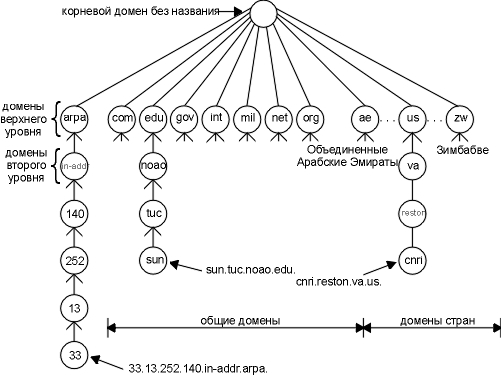
На сетевом уровне сегмент инкапсулируется в **пакет** данных, заголовок которого содержит, кроме прочего, сетевые (логические) адреса отправителя информации (источника) — *Source Address* (**SA**) и получателя (назначения) — *Destination Address* (**DA**). На нашем курсе в качестве DA выступают IP-адреса.

На канальном уровне пакет инкапсулируется в **кадр** или **фрейм** данных, заголовок которого содержит *физические адреса* узла передатчика и приемника и прочую информацию. Кроме того, на этом уровне добавляется **трейлер** (концевик) кадра, содержащий информацию, необходимую для проверки правильности принятой информации. Таким образом, происходит обрамление данных заголовками со служебной информацией, т. е. **инкапсуляция** данных.

**Domain Name System** — это сервис для преобразования доменных имен в IP адреса. Система DNS это, по сути, сеть в сети: если один DNS сервер не знает правильного IP, то он спрашивает у другого и так далее, пока не получит ответ. Распределенная база данных DNS поддерживается с помощью иерархии DNS-серверов, взаимодействующих по определённому протоколу.

**Домен** — узел в дереве имен вместе со всеми подчиненными ему узлами (если таковые имеются), то есть именованная ветвь или поддерево в дереве имен. Структура доменного имени отражает порядок следования узлов в иерархии; доменное имя читается слева направо от младших доменов к доменам высшего уровня (в порядке повышения значимости): вверху находится корневой домен (имеющий идентификатор «.»), ниже идут домены первого уровня (доменные зоны), затем — домены второго уровня, третьего и т. д.

Например, для адреса [1T Спринт](http://sprint.1t.ru/): <https://sprint.1t.ru/> домен первого уровня — ru, второго — 1t, третьего — sprint.



**Поддомен** (англ. subdomain) — подчинённый домен (например, [Wikipedia](http://wikipedia.org/) — поддомен домена org, а [Заглавная страница](http://ru.wikipedia.org/) — поддомен домена [Wikipedia](http://wikipedia.org/)).

**DNS-сервер** — специализированное ПО для обслуживания DNS, а также компьютер, на котором это ПО выполняется. DNS-сервер может быть ответственным за некоторые зоны и/или может перенаправлять запросы вышестоящим серверам.

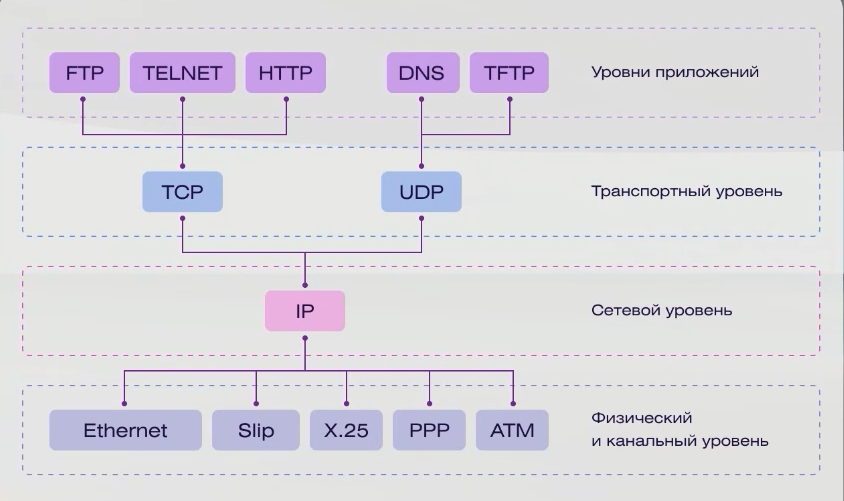
**DNS-клиент** — специализированная библиотека (или программа) для работы с DNS. В ряде случаев DNS-сервер выступает в роли DNS-клиента.

**DNS-запрос** (англ. DNS query) — запрос от клиента (или сервера) серверу. Запрос может быть рекурсивным или нерекурсивным.

:info:Сетевой протокол — это набор правил, определяющий принципы взаимодействия устройств в сети. Чтобы отправка и получение информации прошли успешно, все устройства-участники процесса должны принимать условия протокола и следовать им.

В сети их поддержка встраивается или в аппаратную часть (в «железо»), или в программную часть (в код системы), или и туда, и туда.

# Виды протоколов



В мире существует более 7000 протоколов, и их число продолжает расти. Рассказываем о самых часто используемых правилах взаимодействия устройств в сети.

1. **IP — Internet Protocol**

Протокол передачи, который первым объединил отдельные компьютеры в единую сеть. Этот протокол — самый примитивный в списке. Он ненадежен, поскольку не подтверждает доставку пакетов получателю и не контролирует целостность данных. По протоколу IP передача данных осуществляется без установки соединения.

Основная задача этого протокола — маршрутизация датаграмм, то есть определение пути следования данных по узлам сети.

1. **TCP/IP — Transmission Control Protocol/Internet Protocol**

Это стек протоколов TCP и IP. TCP обеспечивает и контролирует надежную передачу данных и следит за ее целостностью. Он часто используется другими протоколами, состоящими из двух и более протоколов. IP отвечает за маршрутизацию для отправки данных.

1. **UDP — User Datagram Protocol**

Протокол, обеспечивающий передачу данных без предварительного создания соединения между ними. Этот протокол относится к ненадежным. В нем пакеты могут не дойти, прийти не по порядку или вовсе продублироваться.

Основное преимущество UDP протокола заключается в скорости доставки данных. Именно поэтому чувствительные к сетевым задержкам приложения часто используют этот тип передачи данных.

1. **FTP — File Transfer Protocol**

Протокол передачи файлов. Его использовали ещё в 1971 году — задолго до появления протокола IP. На текущий момент этим протоколом пользуются при удалённом доступе к хостингам. FTP является надежным протоколом, поскольку гарантирует передачу данных.

Этот протокол работает по принципу клиент-серверной архитектуры. Пользователь проходит аутентификацию (хотя в отдельных случаях может подключаться анонимно) и получает доступ к файловой системе сервера.

1. **DNS**

Это одновременно и система доменных имён (Domain Name System), и протокол, без которого эта система не смогла бы работать. Протокол позволяет клиентским компьютерам запрашивать у DNS-сервера IP-адрес какого-либо сайта, а также помогает обмениваться базами данных между серверами DNS. В работе этого протокола также используются TCP и UDP.

1. **HTTP — HyperText Transfer Protocol**

Изначально этот протокол использовался для передачи HTML-документов. Сейчас он отвечает за передачу произвольных данных в интернете. Он является протоколом клиент-серверного взаимодействия без сохранения промежуточного состояния. В роли клиента чаще всего выступает веб-браузер, хотя клиентом может быть, например, поисковый робот. Для обмена информацией протокол HTTP в большинстве случаев использует TCP/IP.

HTTP имеет расширение HTTPS, которое поддерживает шифрование. Данные в нём передаются поверх криптографического протокола TLS.

# Общая архитектура приложений

Традиционно веб-сайт — просто комбинация статических страниц. Веб-сайт становится веб-приложением, когда помимо статических в нем появляются еще и динамические страницы.

Архитектура веб-приложений — это механизм, который дает нам представление о том, как устанавливается соединение между клиентом и сервером. Он определяет, как компоненты в приложении взаимодействуют друг с другом. Не имеет значения, какой размер и уровень сложности приложения, все они следуют одному и тому же принципу, отличаясь только деталями.

С технической точки зрения, когда пользователь делает запрос на веб-сайте, взаимодействовать между собой начинают все компоненты приложений, пользовательских интерфейсов, систем промежуточного программного обеспечения, баз данных, серверов и браузера.

:info:Архитектура веб-приложений — структура, которая связывает эти отношения вместе и поддерживает взаимодействие между этими компонентами.

Между запросом пользователя и получением ответа со стороны сервера веб-сайта проходит несколько секунд. За это время браузер успевает получить особый код. Этот код может содержать инструкции, указывающие браузеру на то, как реагировать на различные типы пользовательских вводов.

Вот почему архитектура веб-приложения включает в себя все подкомпоненты и взаимозаменяемые внешние приложения для всего программного приложения — ведь из-за большого объема глобального сетевого трафика она должна обеспечивать надежность, масштабируемость, безопасность и надежность. За годы развития ПО разработчикам удалось придумать надежные подходы, чтобы устранить недостатки проектирования без архитектуры.

Вот некоторые известные архитектуры:

* Многослойная архитектура (Layered Architecture).
* Многоуровневая архитектура (Tiered Architecture).
* Сервис-ориентированная архитектура (Service Oriented Architecture — SOA).
* Микросервисная архитектура (Microservice Architecture).

Рассмотрим самую распространенную трехуровневую систему на примере веб-приложения.

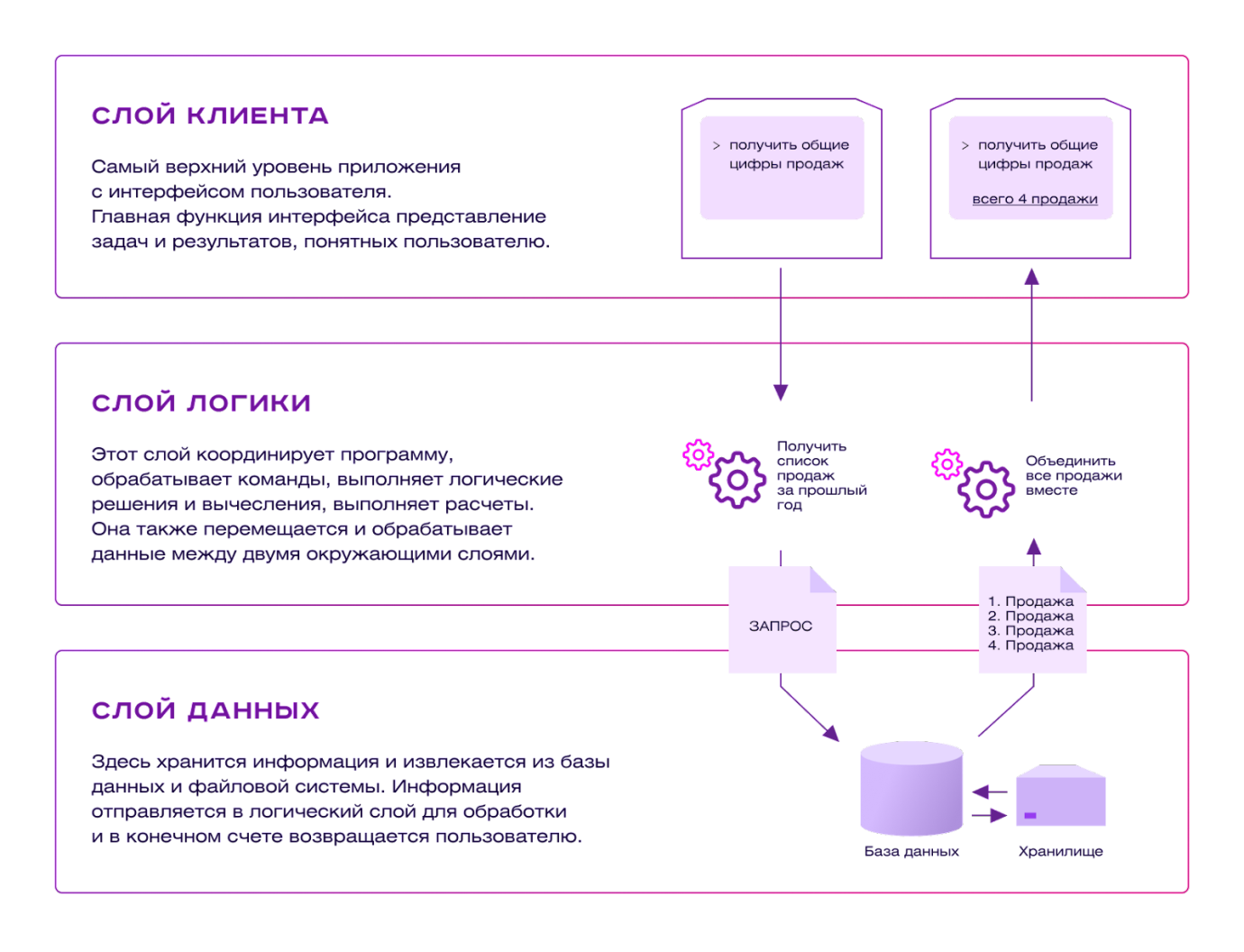
В компьютерных технологиях трехуровневая архитектура (иногда используется слово «трехзвенная») предполагает наличие следующих компонентов приложения:

1. Клиентское приложение (обычно говорят «тонкий клиент» или терминал, в нашем случае браузер).
2. Подключение «тонкого клиента» к серверу приложений (nginx, Apache).
3. Подключение сервера приложений к серверу базы данных (PostgreSQL, MySQL или MS SQL).



Преимущества использования трехуровневой архитектуры включают улучшенную масштабируемость, производительность и доступность.

Подход к разработке приложений с тремя уровнями дает возможность работать над проектом сразу нескольким командам программистов, кодирующих к тому же на разных языках. При этом каждая из команд не зависит от остальных, которые занимаются созданием другого уровня.



Поскольку программный код на каждом уровне может создаваться независимо от остальных уровней, трехуровневая модель облегчает непрерывное развитие приложения для предприятия или программного пакета, по мере появления новых потребностей и возможностей. Существующие приложения могут сохраняться и инкапсулироваться в новый уровень, компонентом которого они становятся.

Все приложения состоят из следующих основных компонентов:

1. Сторона клиента

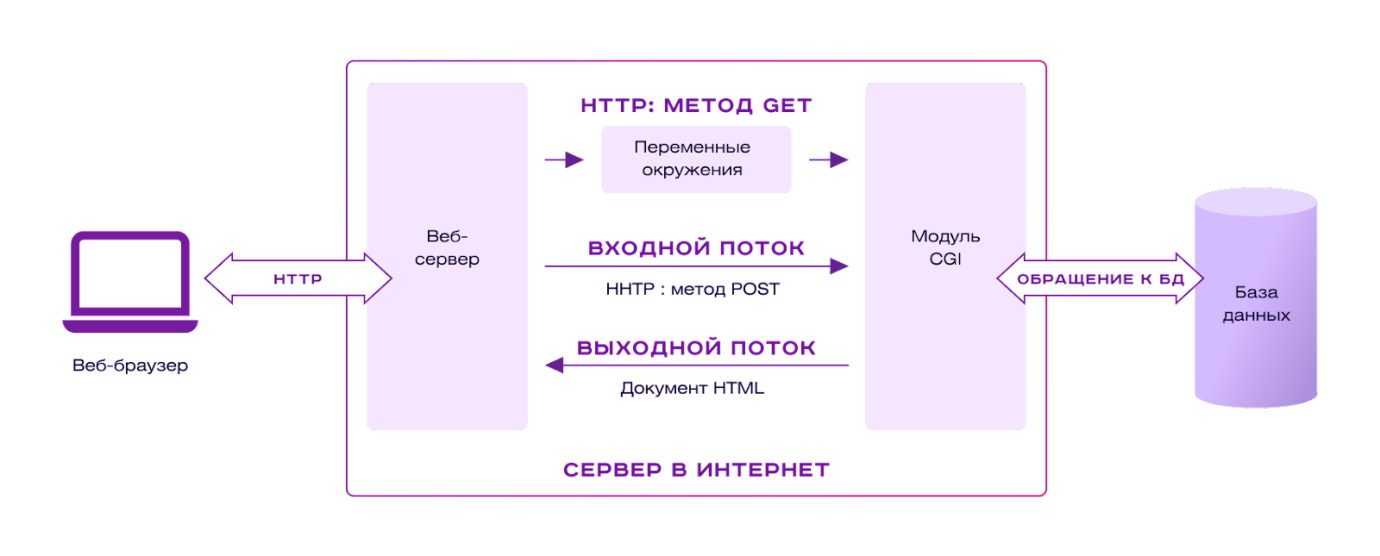
Сторона клиента более известна как интерфейс. Код для нее пишут на HTML, CSS, JavaScript и загружают в браузере. Именно здесь происходит взаимодействие с пользователем.

1. Серверная часть

Серверная часть, иначе бэкэнд, управляет бизнес-логикой и отвечает на HTTP-запросы. Серверный код пишут, как правило, на языках Java и PHP.

1. Сервер базы данных

Это дополнительный компонент. Он отправляет запрошенные данные на сторону сервера.

****

# Дополнительный материал

## 1 Информационные технологии

Информационные технологии можно разделить на основе информационных потоков:

* на технологии, в которых преобладает смысловое содержание (происходит преобразование информации в данные и наоборот);
* на технологии, работающие с данными, в которых преобладает синтаксический аспект информации;
* на технологии, работающие со знаниями, в которых преобладает семантический аспект информации.

Информационные технологии характеризуются следующими основными свойствами:

* предметом (объектом) обработки (процесса) являются данные;
* целью процесса является получение информации;
* средствами осуществления процесса являются программные, аппаратные и программно-аппаратные вычислительные комплексы;
* процессы обработки данных разделяются на операции в соответствии с данной предметной областью;
* выбор управляющих воздействий на процессы должен осуществляться лицами, принимающими решение;
* критериями оптимизации процесса являются своевременность доставки информации пользователю, ее надежность, достоверность, полнота.

## 2 Информация: структурирование данных

Структурирование данных — это процесс, приводящий к определенной форме записи данных об объектах одного класса. К примеру объектом могут быть студенты университета, т.е. Класс данных «Студент», объект «Василий Петрович Андреев», данные всех студентов приводятся к единой форме (Анкета содержащая ФИО, факультет, курс).

Предметная область — это совокупность объектов, которые находятся между собой в определенных отношениях и связях. В нашем случае это будет Университет, в рамках которого присутствуют следующие объекты: Студенты, Преподаватели, Специальности, Пары (расписание), Экзамены и т.д.

Реляционная база данных — база данных, основанная на реляционной модели данных. Понятие «реляционный» основано на англ. relation («отношение, зависимость, связь»).

Использование реляционных баз данных было предложено доктором Коддом из компании IBM в 1970 году. Для работы с реляционными БД применяют реляционные СУБД.

Модель данных — это совокупность взаимосвязанных по определенному правилу данных.

Реляционная модель данных представляет собой совокупность таблиц с установленными связями.

Свойства реляционной модели данных:

* Каждый элемент таблицы — один элемент данных.
* Все элементы одного столбца (поля) имеют одинаковый тип (числовой, символьный и т.п.), формат и смысл.
* Каждый столбец (поле) имеет уникальное имя.
* Одинаковые строки в таблице отсутствуют.
* Порядок следования строк в таблице может быть произвольным.
* Каждая таблица должна иметь ключ. Ключ — это поле, которое единственным образом определяет каждую строку(запись) в таблице.
* Таблицы, входящие в модель данных, могут характеризоваться разным количеством полей и записей.

## 3 Сети, протоколы, пространство имен

В Интернете существует единая система адресации, которая помогает компьютерам найти друг друга в процессе обмена информацией. Интернет работает по протоколу TCP/IP (протокол, который предназначен для управления передачей данных интернета, выполняет функции транспортного уровня модели OSI). Протокол – это правило передачи информации в Сети.

Передача данных происходит через уникальный IP-адрес. IP-протокол – это набор правил, который позволяет доставить данные от одного компьютера к другому благодаря знанию IP-адресов отправителя и получателя.

IP-адреса бывают 2 стандартов:

* IPv4, цифровые адреса в десятичной системе из 4 блоков до 3 цифр через точки, например, 192.168.12.33, эти адреса уже в мире почти закончились и стоят все дороже, но на данный момент мы в основном оперируем ими, и они есть у всех наших площадок.
* IPv6, цифровые адреса в шестнадцатеричной системе из 8 блоков до 4 символов через двоеточия, например, 1050:0:0:0:5:600:300c:326b, эти адреса раздаются практически бесплатно, и в будущем полностью заменят IPv4, но мы их пока не особо используем, в виду определенных технологических сложностей.

Простыми словами Интернет состоит из доменов, адресации, хостингов (серверов) и информационных систем (содержат логику и данные).

*Домены*

Домен / доменное имя / имя – это уникальное название сайта (как номерной знак у автомобиля). Его пользователи вводят в браузере для перехода к сайту. Например, google.com, 1t.vc или sprint.1t.ru. Пишется как в латиннице, так и есть домены в кириллице. Его можно взять в аренду или отказаться от него. В обиходе слово “аренда” домена заменяется словом “покупка”. Арендодатель доменов – это регистратор доменов. По имени домена открывается сайт до тех пор, пока за него платит арендатор. Расчет за домен происходит по предоплате ежемесячной или ежегодной. После окончания оплаты домен еще какое-то время хранится в резерве (1-2 месяца), а затем регистратор домена опять выставляет его на продажу.

Домены бывают нескольких уровней, разделенных точкой. Порядок считается справа налево.

Домены первого (верхнего) уровня com, vc, run и т.д. – это международные или национальные домены. Они не продаются. Это как в номерных знаках автомобилей вначале указывается флаг страны и сокращенное название страны, выдавшей номера, в латиннице.

Домены второго уровня, вида 1t.ru, можно купить. Это именно то, что нам нужно для наших систем, и именно связку домена первого и второго уровней обычно и называют “домен”. Подобрать домен для покупки самостоятельно можно на сайтах различных регистраторов доменов, например, nic.ru, reg.ru и др.

Подбор домена — это процедура определения доступности выбранного вами имени домена, вот по ссылкам <https://www.nic.ru/catalog/domains/>и <https://www.reg.ru/domain/new/>можно указать домен второго уровня, и сервис покажет какие домены свободны, а какие заняты. Все регистраторы доменов работают с общей мировой базой доменов, поэтому при покупке лучше доверять проверенным продавцам.

При необходимости домен можно перенести от одного регистратора доменов к другому. Также домены в разных зонах первого уровня имеют различную стоимость, т.е. цены в зонах com и ru отличаются. Купленный домен второго уровня зачастую мы направляем на наши сервера управления доменами (DNS-сервера), а иногда пользуемся панелью управления доменом от регистратора. Бывает, что заказчик уже имеет свой домен и хочет его использовать для системы, в таком случае он должен направить свой этот домен на наши DNS-сервера, или передать нам доступ к панели управления своим доменом, или, если он достаточно компетентен, то самостоятельно выполнить настройку домена, как мы это ему скажем.

Под настройкой домена обычно подразумевается направление домена на хостинг площадки (А-записи) и/или почтовые сервера (MX-записи).

Купив домен второго уровня, мы уже сами можем создавать в нем домены третьего уровня без дополнительных оплат, например, sprint.1t.ru. Создание домена третьего уровня обычно называют созданием “поддомена”. Т.е. для нашего домена 1t.ru при необходимости, мы можем смело создать практически неограниченного количество поддоменов, например, webinar.1t.ru, project.1t.ru и т.д.

При этом создание поддомена www также является обязательным и выполняется сразу при покупке. Если этого не сделать, то по адресу www.1t.ru сайт не откроется.

*DNS-сервер*

Управление доменом происходит через DNS-сервер (Name-сервер, nameserver, NS) — сервер, преобразующий доменные имена, с которыми работают пользователи, в понятные компьютерам IP-адреса или в обратном направлении. Т.е. у каждого сервера в Интернете есть свой уникальный сложный IP-адрес, и вот именно DNS-сервер выполняет преобразование и направление запроса пользователя по доменному имени по маршруту к целевому IP-адресу сервера с хостингом, который может находиться в любой точке планеты.

*Хостинг*

Хостинг – это место, где хранятся исходные коды системы, ее база данных (БД), файлы системы и пользователей и т.д. и которое имеет IP-адрес.

Связку “хостинг”+”домен” часто заменяем словом “площадка”. В зависимости от масштабов системы выбирается ее хостинг. Хостинг может занимать как весь физический сервер, так и может быть, что на одном сервере расположено множество разделенных площадок. К выбору хостинга нужно подходить ответственно, чтобы площадка не “упала” под нагрузкой. При планировании выбора хостинга нужно отталкиваться от следующих параметров:

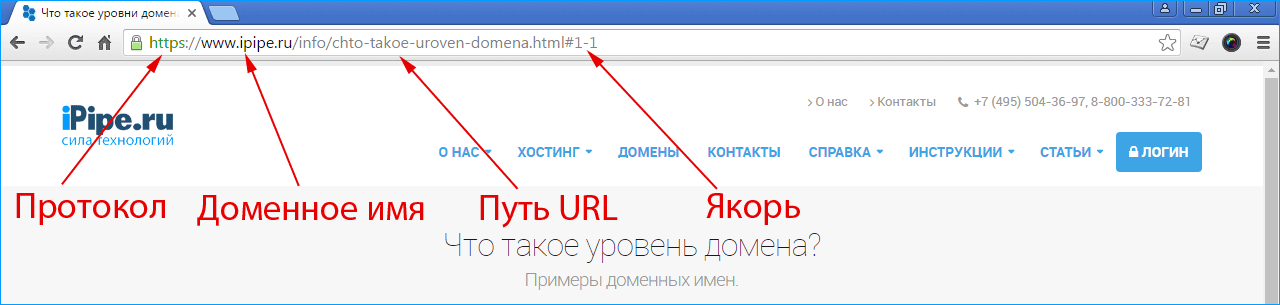
* планируемое количество пользователей системы в онлайне,
* размер БД и хранилища файлов,
* особенности контента сайта (будут ли пользователи скачивать/просматривать много видео).

Как уже было сказано выше, одному хостингу (площадке) может быть присвоено множество доменных имен. Например, может быть yandex.ru и ya.ru. По каждому из этих алиасов открывается одна и та же система.

Например, для сайта www.site.ru может быть имя кириллицей сайт.рф. Эти дополнительные домены называются “алиасами” (псевдонимами) сайта. И их можно в любое время подключать и отключать от площадок.

Алиас сайта (от англ. Alias – псевдоним, синоним) – это альтернативное доменное имя. Например, когда пользователи вводят в адресную строку домен-синоним, на экране отображается тот же по содержанию веб-ресурс, который доступен и по оригинальной ссылке, но под другим URL.

Для обращения к сайту на хостинге используется URL (Uniform Resource Locator).



URL-адрес, который мы видим в адресной строке браузера, состоит из нескольких частей.

* В начале адреса (1) всегда указан протокол (в некоторых браузерах по умолчанию он может быть скрыт и становится виден при щелчке по адресной строке). Если мы просматриваем веб-страницу, это будет протокол передачи данных «http» или «https».
* После протокола (2) следует доменное имя сайта (в редких случаях его IP-адрес, а также в некоторых случаях URL-адрес может содержать номер порта через двоеточие, например, http://example.com:8080).
* Затем указывается путь к странице (3), состоящий из каталогов и подкаталогов. Если путь в урле не указан, то открывается страница системы, настроенная по умолчанию. URL также может включать параметры, которые указываются после знака «?» и разделяются символом «&».
* Конечный компонент URL, который пользователь может увидеть в документах большого объема, состоящих из нескольких разделов, — это якорь, которому предшествует знак решетки «#» (4). Часть адреса после этого знака ссылается на определенный блок внутри страницы сайта.