**Рефлексия** - API, позволяющих анализировать приложение в момент исполнения

**Интерфейс** — определение набора методов, которые должен реализовывать объект.

Если объект содержит реализацию указанных методов, то говорят, что **объект реализует интерфейс**.

Изначально, интерфейс — **это чистая абстракция**: мы определяем набор методов, которые должны быть в объекте, а потом используем этот интерфейс в качестве типа данных

(дляпеременных, параметров и полей).



# List — наследник Collection. Он определяет упорядоченную коллекцию элементов, в которой есть понятие индекса (т.е. порядкового

# номера элемента в коллекции):

# SET

Set — это коллекция, не содержащая повторяющихся элементов.Повторяются элементы или нет, определяется через equals.Коллекция не вводит понятия индекса, поэтому, методов для работы с индексами не предусмотрено.

Готовые реализации: *HashSet TreeSet*

# MAP

Map — коллекция, содержащая пары ключ-значение, в которых ключи являются уникальными. Уникальность ключей определяется через equals. Важно: интерфейс Map не является наследником Collection.

Готовые реализации: *HashMap TreeMap*

# ТЕСТИРОВАНИЯ CRUD систем.

# 1. Проверяем систему в трёх состояниях:

1. Пустая система
2. В системе один объект

c. В системе больше одного объекта

1. **Проверяем работу с существующими и несуществующими объектами** (получение по существующему/несуществующему id, удаление по id)

# Проверяем поиск на трёх возможных возвращаемых значениях:

* 1. Ничего не найдено
  2. Найден ровно один объект
  3. Найдено несколько объектов (тогда смотрим на их порядок)

# Ключевой вопрос: как быть с добавлением?

Ведь чтобы получить состояние системы с одним или несколькими элементами, в систему нужно эти элементы добавить.

Существующие варианты:

1. Вынести добавление в «преднастройку системы» (как мы и делали в некоторых ДЗ).
2. Сделать «вспомогательный» метод или конструктор, который позволяет создавать систему с нужным количеством объектов.

# CRUD — ПРЕДНАСТРОЙКА СИСТЕМЫ

В случае с преднастройкой системы обычно success-поведение самого метода добавления не тестируют, т.к. если он будет работать некорректно, тесты, которые от него зависят

«попадают».

При этом писать тесты на генерируемые исключения и бизнес-логику в самом методе добавления нужно.

# CRUD — ВСПОМОГАТЕЛЬНЫЕ МЕТОДЫ

В случае со вспомогательными методами или конструкторами необходимо писать все тесты на метод добавления и следить за тем, чтобы вспомогательные методы и

конструкторы не использовались «не тестами», либо покрывать тестами и их тож

Таким образом, как всегда — идеального решения не существует, вы должны выбирать и

учитывать недостатки обоих способов.

Здесь (как всегда) всё зависит от стиля принятого в определённой команде. Два самых распространённых\*:

1. В название класса теста добавляете название состояния: EmptyProductRepositoryTest, SingleItemProductRepositoryTest (и другие вариации).
2. Внутри класса теста делаете вложенные классы, описывающие состояние, и уже в них тесты (см.следующий слайд).

Примечание\*: конечно, есть и другие вариации (например, с классами,вложенными друг в друга матрёшкой



**class:**создание нового класса/модели.  
**method:** функция внутри класса.  
**constructor:**метод, который инициирует объект при создании экземпляра класса.  
**extends:** используется для определения наследования.  
**super:**метод, который устанавливает свойства наследования за счет вызова родительского конструктора. Метод super должен стоять первой строкой в методе constructor.  
**get:** возвращает значение.  
**set:** переопределяет новое существующее значение.  
**new:** создание объекта через метод конструктора класса

**SQL Injection** - подмена или вставка sql запроса в базу данных

**XSS-атака** - замена кода в сайте с целью обмана пользователя,подмена сайта целиком

**Code Injection** - Inject code in system/program

**Fiddler** - Traffic packages manager

**Wireshark** - TRAFFIC Manager

Настройка Android Studio - Запуск проекта, интерфейс на основе intellij, есть декомпиляция apk файлов.

**Тестирование UI** - тестирование интерфейса пользователя - тестирование функциональности приложения со стороны пользователя

**Автоматизация тестов с помощью фреймворка Espresso** - Тестирование ui через Android Studio

**Автоматизация с использованием Appium** - эмуляция сервера для android приложений во время разработки

**Javascript** - высокоуровневый язык для веб инфраструктуры, Исполняет запросы пользователя на клиенсткой машине

**Отладка,** **Developer Tools** - коммандная панель браузера которая дает разработчику возможности мененджмента страницы

**Типы данных в JavaScript**

В JavaScript есть 8 основных типов.

number для любых чисел: целочисленных или чисел с плавающей точкой; целочисленные значения ограничены диапазоном ±(253-1).

bigint для целых чисел произвольной длины.

string для строк. Строка может содержать ноль или больше символов, нет отдельного символьного типа.

boolean для true/false.

null для неизвестных значений – отдельный тип, имеющий одно значение null.

undefined для неприсвоенных значений – отдельный тип, имеющий одно значение undefined.

object для более сложных структур данных.

symbol для уникальных идентификаторов.

Оператор typeof позволяет нам увидеть, какой тип данных сохранён в переменной.

Имеет две формы: typeof x или typeof(x).

Возвращает строку с именем типа. Например, "string".

Для null возвращается "object" – это ошибка в языке, на самом деле это не объект.

**(npm)** - developer package manager with command line

**Асинхронность** - Это прием программирования, который использует многпоточность, выполняя несколько операций одновременно, во избежание блокировки.

**Платформа Node.js** - платформа расширяющая возможности javascript в серверах и в железе

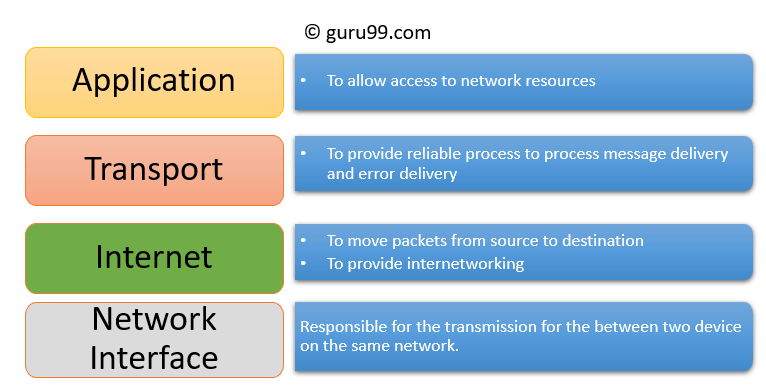
**Jest** - testing network for Javascript projects (requers nvm/node js)

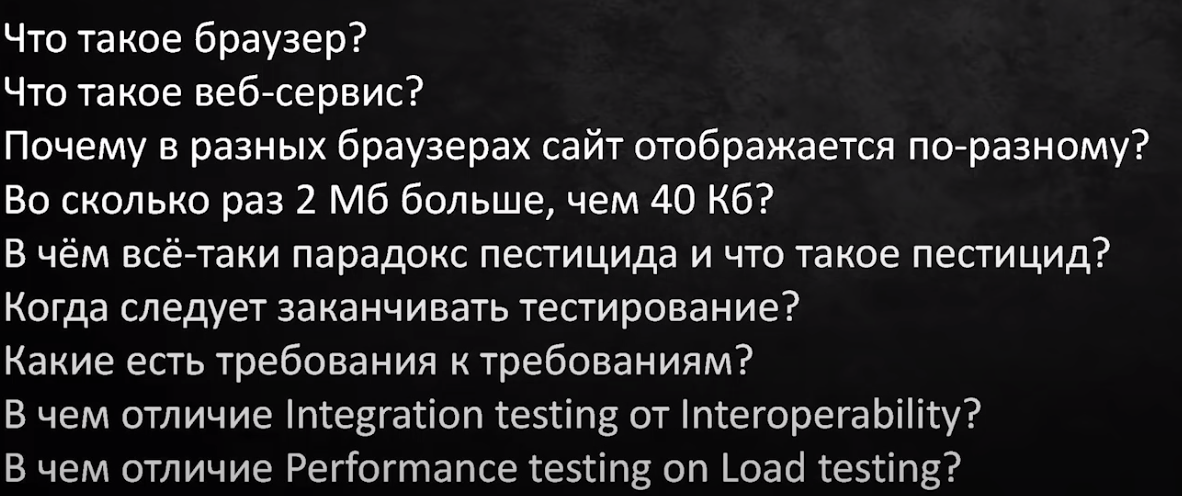
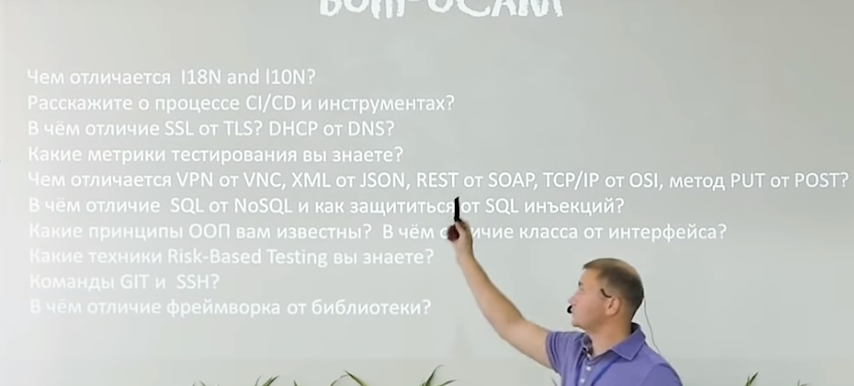
**Puppeeter** - testing framework for Chromium

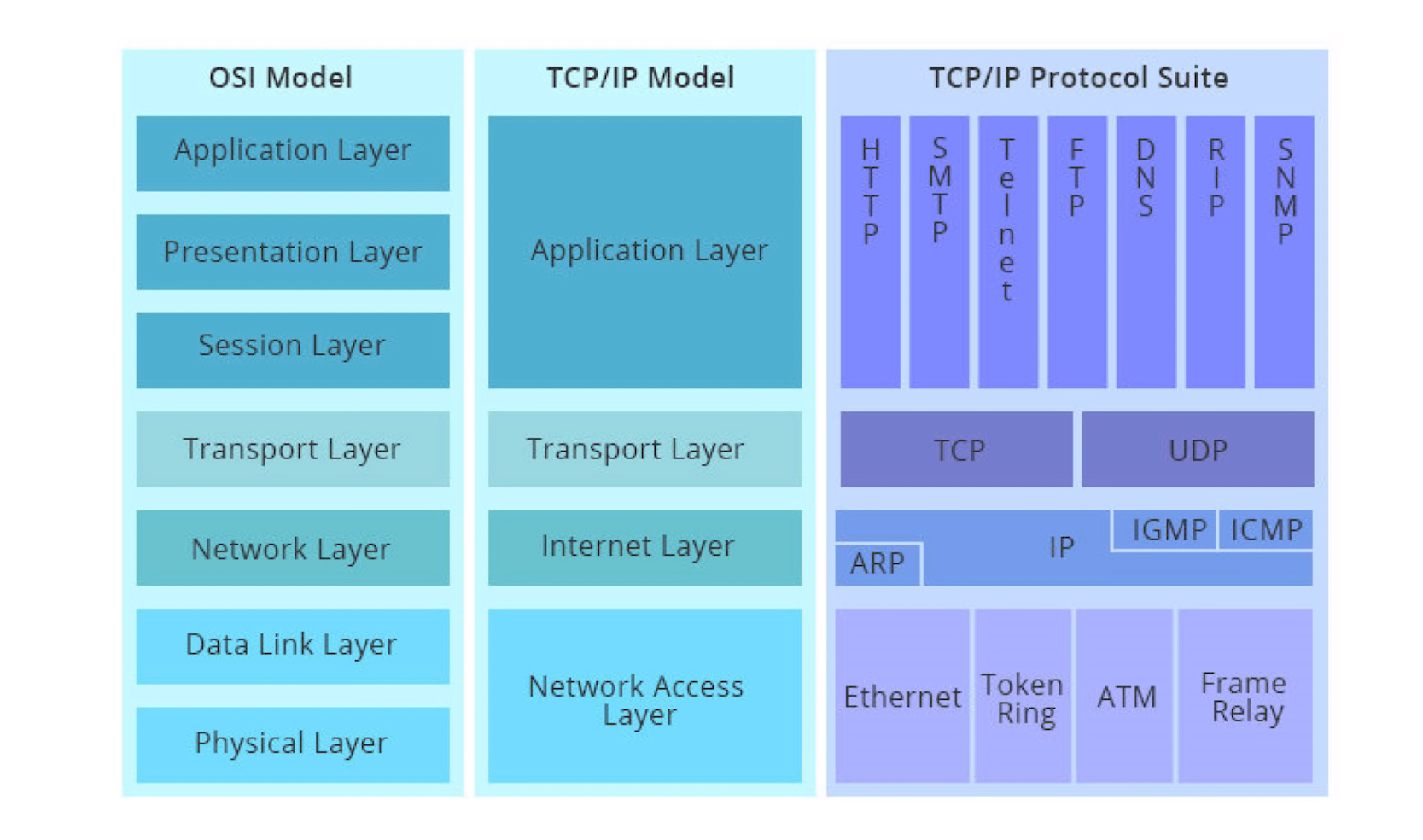
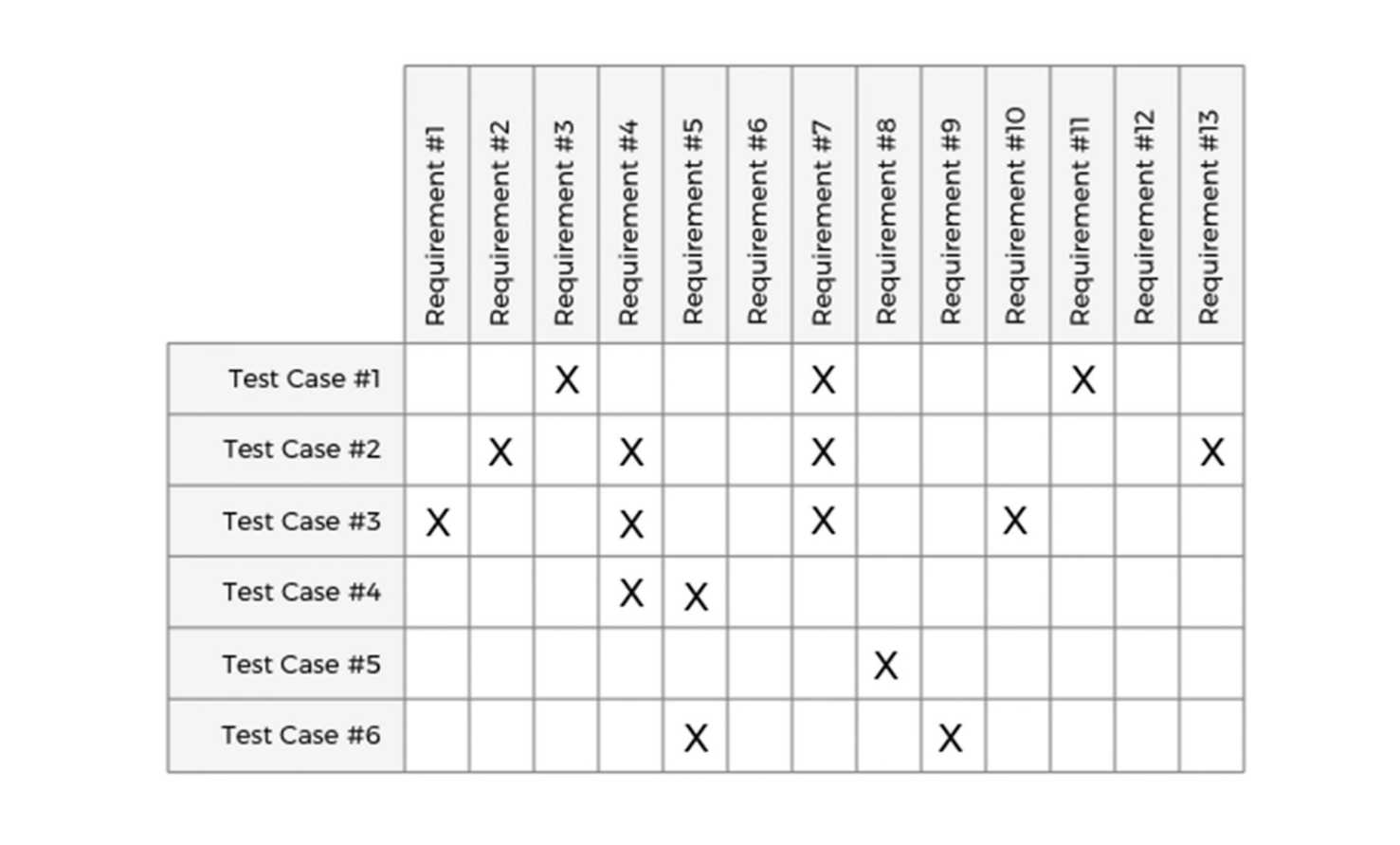
**Playwright** - Testing framork for Chrome,Firefox and Safari

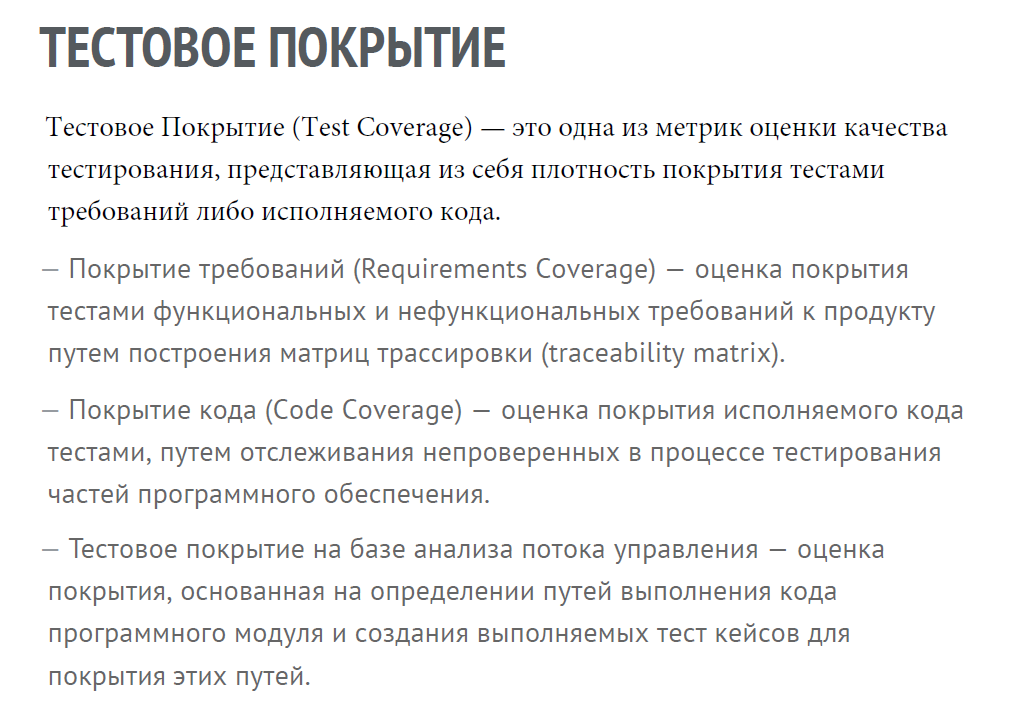
**Cypress** - End to End Javascrypt Testing tool

**TCP/IP** — [сетевая модель](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) передачи данных, представленных в цифровом виде. Модель описывает способ передачи данных от источника информации к получателю. В модели предполагается прохождение информации через четыре уровня, каждый из которых описывается правилом (протоколом передачи). Наборы правил, решающих задачу по передаче данных, составляют стек [протоколов передачи данных](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), на которых базируется [Интернет](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82)[[1]](https://ru.wikipedia.org/wiki/TCP/IP#cite_note-oslogic-1)[[2](https://ru.wikipedia.org/wiki/TCP/IP#cite_note-ciscolearning-2)

**Уровни TCP/IP**

**Основой HTTP** является [технология «клиент-сервер»](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) Обмен сообщениями идёт по обыкновенной схеме «запрос-ответ». Для идентификации ресурсов HTTP использует глобальные [URI](https://ru.wikipedia.org/wiki/URI). В отличие от многих других протоколов, HTTP не сохраняет своего состояния.



Сетевая модель OSI (The Open Systems Interconnection model) — сетевая модель стека (магазина) сетевых протоколов OSI/ISO. Посредством данной модели различные сетевые устройства могут взаимодействовать друг с другом. Модель определяет различные уровни взаимодействия систем. Каждый уровень выполняет определённые функции при таком взаимодействии.

7. The application layer

This is the only layer that directly interacts with data from the user. Software applications like web browsers and email clients rely on the application layer to initiate communications. But it should be made clear that client software applications are not part of the application layer; rather the application layer is responsible for the protocols and data manipulation that the software relies on to present meaningful data to the user. Application layer protocols include HTTP as well as SMTP (Simple Mail Transfer Protocol is one of the protocols that enables email communications).

The Presentation Layer

6. The presentation layer

This layer is primarily responsible for preparing data so that it can be used by the application layer; in other words, layer 6 makes the data presentable for applications to consume. The presentation layer is responsible for translation, encryption, and compression of data.

Two communicating devices communicating may be using different encoding methods, so layer 6 is responsible for translating incoming data into a syntax that the application layer of the receiving device can understand.

If the devices are communicating over an encrypted connection, layer 6 is responsible for adding the encryption on the sender’s end as well as decoding the encryption on the receiver’s end so that it can present the application layer with unencrypted, readable data.

Finally the presentation layer is also responsible for compressing data it receives from the application layer before delivering it to layer 5. This helps improve the speed and efficiency of communication by minimizing the amount of data that will be transferred.

The Session Layer

1. The session layer

This is the layer responsible for opening and closing communication between the two devices. The time between when the communication is opened and closed is known as the session. The session layer ensures that the session stays open long enough to transfer all the data being exchanged, and then promptly closes the session in order to avoid wasting resources.

The session layer also synchronizes data transfer with checkpoints. For example, if a 100 megabyte file is being transferred, the session layer could set a checkpoint every 5 megabytes. In the case of a disconnect or a crash after 52 megabytes have been transferred, the session could be resumed from the last checkpoint, meaning only 50 more megabytes of data need to be transferred. Without the checkpoints, the entire transfer would have to begin again from scratch.

The Transport Layer

1. The transport layer

Layer 4 is responsible for end-to-end communication between the two devices. This includes taking data from the session layer and breaking it up into chunks called segments before sending it to layer 3. The transport layer on the receiving device is responsible for reassembling the segments into data the session layer can consume.

The transport layer is also responsible for flow control and error control. Flow control determines an optimal speed of transmission to ensure that a sender with a fast connection doesn’t overwhelm a receiver with a slow connection. The transport layer performs error control on the receiving end by ensuring that the data received is complete, and requesting a retransmission if it isn’t.

The Network Layer

1. The network layer

The network layer is responsible for facilitating data transfer between two different networks. If the two devices communicating are on the same network, then the network layer is unnecessary. The network layer breaks up segments from the transport layer into smaller units, called packets, on the sender’s device, and reassembling these packets on the receiving device. The network layer also finds the best physical path for the data to reach its destination; this is known as routing.

The Data Link Layer

1. The data link layer

The data link layer is very similar to the network layer, except the data link layer facilitates data transfer between two devices on the SAME network. The data link layer takes packets from the network layer and breaks them into smaller pieces called frames. Like the network layer, the data link layer is also responsible for flow control and error control in intra-network communication (The transport layer only does flow control and error control for inter-network communications).

The Physical Layer

1. The physical layer

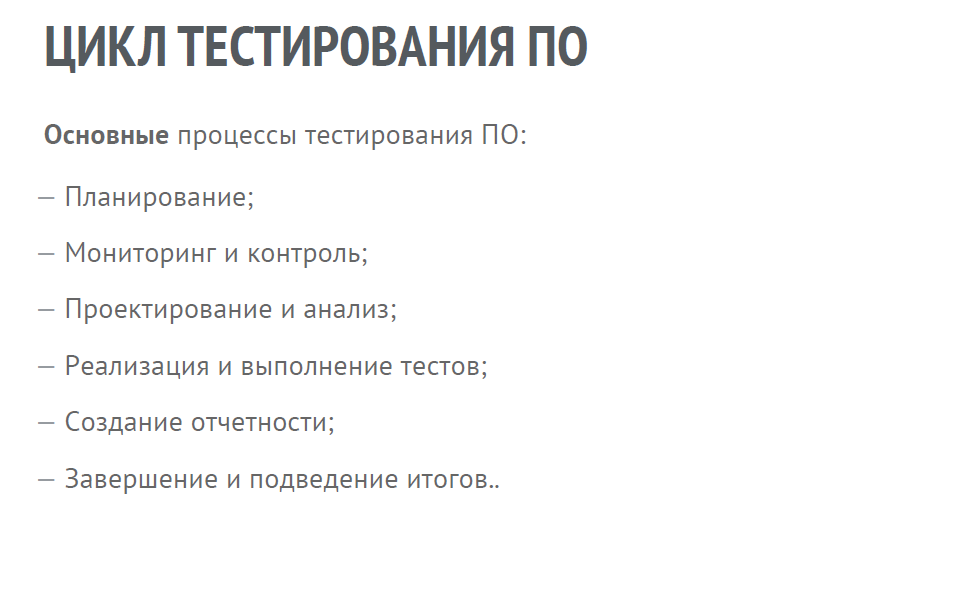
This layer includes the physical equipment involved in the data transfer, such as the cables and switches. This is also the layer where the data gets converted into a bit stream, which is a string of 1s and 0s. The physical layer of both devices must also agree on a signal convention so that the 1s can be distinguished from the 0s on both devices.

How data flows through the OSI Model

In order for human-readable information to be transferred over a network from one device to another, the data must travel down the seven layers of the OSI Model on the sending device and then travel up the seven layers on the receiving end.

For example: Mr. Cooper wants to send Ms. Palmer an email. Mr. Cooper composes his message in an email application on his laptop and then hits ‘send’. His email application will pass his email message over to the application layer, which will pick a protocol (SMTP) and pass the data along to the presentation layer. The presentation layer will then compress the data and then it will hit the session layer, which will initialize the communication session.

The data will then hit the sender’s transportation layer where it will be segmented, then those segments will be broken up into packets at the network layer, which will be broken down even further into frames at the data link layer. The data link layer will then deliver those frames to the physical layer, which will convert the data into a bitstream of 1s and 0s and send it through a physical medium, such as a cable.

Once Ms. Palmer’s computer receives the bit stream through a physical medium (such as her wifi), the data will flow through the same series of layers on her device, but in the opposite order. First the physical layer will convert the bitstream from 1s and 0s into frames that get passed to the data link layer. The data link layer will then reassemble the frames into packets for the network layer. The network layer will then make segments out of the packets for the transport layer, which will reassemble the segments into one piece of data. 

The data will then flow into the receiver’s session layer, which will pass the data along to the presentation layer and then end the communication session. The presentation layer will then remove the compression and pass the raw data up to the application layer. The application layer will then feed the human-readable data along to Ms. Palmer’s email software, which will allow her to read Mr. Cooper’s email on her laptop screen

The User Datagram Protocol (**UDP**) is a lightweight data transport protocol that works on top of IP. **UDP** provides a mechanism to detect corrupt data in packets, but it does not attempt to solve other problems that arise with packets, such as lost or out of order packets

FTP - The File Transfer Protocol is a standard communication protocol used for the transfer of computer files from a server to a client on a computer network. FTP is built on a client–server model architecture using separate control and data connections between the client and the server

SMTP -The Simple Mail Transfer Protocol is an internet standard communication protocol for electronic mail transmission. Mail servers and other message transfer agents use SMTP to send and receive mail messages.

Telnet is an application protocol used on the Internet or local area network to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection

**DNS** ([англ.](https://ru.m.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA" \o "Английский язык) *Domain Name System* «система доменных имён») — компьютерная [распределённая система](https://ru.m.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85" \o "Распределённая база данных) для получения информации о [доменах](https://ru.m.wikipedia.org/wiki/%D0%94%D0%BE%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B8%D0%BC%D1%8F" \o "Доменное имя). Чаще всего используется для получения [IP-адреса](https://ru.m.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81" \o "IP-адрес) по имени [хоста](https://ru.m.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82" \o "Хост) (компьютера или устройства), получения информации о маршрутизации почты и/или обслуживающих узлах для протоколов в домене ([SRV-запись](https://ru.m.wikipedia.org/wiki/SRV-%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C" \o "SRV-запись)).

The Routing Information Protocol is one of the oldest distance-vector routing protocols which employs the hop count as a routing metric. RIP prevents routing loops by implementing a limit on the number of hops allowed in a path from source to destination

Simple Network Management Protocol (**SNMP**) is an Internet Standard protocol for collecting and organizing information about managed devices on IP networks and for modifying that information to change device behavior

**Frame relay** is a protocol that defines how **frames** are routed through a fast-packet network based on the address field in the **frame**. ... This support allows a network of systems to communicate using the **frame**-**relay** network as a backbone, without the need for multiple leased T1 lines

**Asynchronous Transfer Mode** (**ATM**) is a [telecommunications](https://en.m.wikipedia.org/wiki/Telecommunications) standard defined by [ANSI](https://en.m.wikipedia.org/wiki/ANSI) and [ITU](https://en.m.wikipedia.org/wiki/ITU) (formerly CCITT) for digital transmission of multiple types of traffic, including [telephony](https://en.m.wikipedia.org/wiki/Telephony) (voice), [data](https://en.m.wikipedia.org/wiki/Data_transmission), and [video](https://en.m.wikipedia.org/wiki/Video) signals in one network without the use of separate overlay networks.[[](https://en.m.wikipedia.org/wiki/Asynchronous_Transfer_Mode#cite_note-1)

**Token Ring** is a computer networking technology used to build local area networks. It was introduced by IBM in 1984, and standardized in 1989 as IEEE 802.5. It uses a special three-byte frame called a **token** that is passed around a logical **ring** of workstations or servers.

Ethernet protocol is a typical LAN technology. Standard Ethernet-based local area networks transmit data at speed up to 10 Mbps. New Ethernet cards known as Fast Ethernet represent high-speed LAN technology as it can provide data transfer rates as high as 100 Mbps

The Address Resolution Protocol is a communication protocol used for discovering the link layer address, such as a MAC address, associated with a given internet layer address, typically an IPv4 address. This mapping is a critical function in the Internet protocol suite

The Internet Group Management Protocol is a communications protocol used by hosts and adjacent routers on IPv4 networks to establish multicast group memberships. IGMP is an integral part of IP multicast and allows the network to direct multicast transmissions only to hosts that have requested them

The Internet Control Message Protocol (ICMP) is a supporting protocol in the Internet protocol suite. It is used by network devices, including routers, to send error messages and operational information indicating success or failure when communicating with another IP address, for example, an error is indicated when a requested service is not available or that a host or router could not be reached.[2] ICMP differs from transport protocols such as TCP and UDP in that it is not typically used to exchange data between systems, nor is it regularly employed by end-user network applications (with the exception of some diagnostic tools like ping and traceroute).

Бра́узер, или веб-обозрева́тель (от англ. Web browser, МФА: [wɛb ˈbraʊ. Zə(ɹ), -zɚ]; устар. Бро́узер) — прикладное программное обеспечение для просмотра страниц, содержания веб-документов, компьютерных файлов и их каталогов; управления веб-приложениями; а также для решения других задач.

The term Web service is either: a service offered by an electronic device to another electronic device, communicating with each other via the World Wide Web, or a server running on a computer device, listening for requests at a particular port over a network, serving web documents

Причина этой проблемы — отличия в исходном коде. Особенно это касается старых версий **браузеров**. Нередко определенные HTML-коды и CSS-стили по-**разному** воспринимаются и обрабатываются на **разных** движках, поэтому один и тот же элемент может в одном браузере **отображаться** и функционировать корректно, а в другом — нет

Во сколько раз 2 Мб больше, чем 40 Кб?

**Используемые сведения**: 1 Мб = 1024 Кб

Рекомендуется записать условие задачи в следующем виде:

**Дано:**

**V1= 2 Мб**

**V2 = 40 Kб**

**Решение:**

**1). V1= 2 Мб = 2 \* 1024 Кб = 2048 Кб**

**2). V1/ V2 = 2048 Кб / 40 Кб = 51,2 **

 51 (раз)

#### Что такое парадокс пестицида?На большинстве проектов случаются такие ситуации, когда чем больше мы тестируем продукт, тем больший иммунитет вырабатывается у багов, которые мы пытаемся найти, используя наши тестовые наборы. Когда одни и те же тесты повторяются снова и снова, то в конце концов они перестают находить новые багиСледует выделить 3 основных **критерия** для **остановки**, завершения **тестирования**: Время Бюджет Все тест кейсы пройдены, найденные баги исправлены и перепроверены

**Всего критериев качества требований можно выделить девять:**

* Корректность Бывает что в требованиях допускают ошибки при составлении таковых. ...
* Недвусмысленность ...
* Полнота ...
* Непротиворечие ...
* Упорядоченность по важности и стабильности ...
* Поддаваемость проверке ...
* Модифицируемость ...
* Понимабельность

**Тестирование взаимодействия** (**Interoperability Testing**) – это функциональное тестирование, проверяющее способность приложения взаимодействовать с одним и более компонентами или системами и включающее в себя тестирование совместимости (compatibility testing) и интеграционное тестирование (**[integration testing](http://www.protesting.ru/testing/levels/integration.html)**).

Тестирование производительности измеряет время отклика приложения с ожидаемым числом пользователей. Целью этого является получение данных о том, что приложение работает в нормальных условиях. Соответствует ли оно требуемому времени отклика?

### Нагрузочное тестирование

Нагрузочное тестирование измеряет время отклика, когда приложение подвергается большему числу пользователей.

Время отклика будет увеличиваться, т. е. приложение будет работать медленнее при большей нагрузке, но целью нагрузочного тестирования является проверка возможности приложения при увеличении нагрузки на сервер приводить к сбоям и уничтожению серверов.

Нагрузочное тестирование обычно начинается с малых чисел и постепенно увеличивается за определенный период времени, пока не достигнет желаемой нагрузки на систему, а затем замедляется.

#### **Интернационализация (I18N)**

Звучит интересно, но как же нам заставить всё это работать именно в таком виде? Тут нам на помощь приходит интернационализация. Интернационализация — процесс разработки приложения, при котором код самого приложения независим от любых языковых и культурных особенностей региона/страны (cultural specific data). Internationalization принято сокращать как "I18N", где 18 — это число символов между "I" и "N". Суть интернационализации в том, чтобы сделать процесс локализации проще, дешевле и быстрее. Реализацию I18N обычно начинают на ранних этапах проекта, чтобы подготовить ваш продукт к будущей локализации.

Локализация (L10N)

Теперь можно поговорить чуть подробнее и о локализации. Локализация (localization или L10N) — это процесс перевода и культурной адаптации продукта к особенностям определенной страны, региона. Как упоминалось выше, на этой стадии участники разработки продукта работают с локалями — внешними ресурсами (файлами), которые подгружаются приложением для загрузки локализации для вашей страны/региона. Нюансов, на которые стоит обратить внимание, очень большое количество и их я приведу в разделе с чеклистом ниже, а пока хочу выделить основные зоны локализации:

Текст и связанные с ним функции (например сортировка, поиск, поддержка спец. Символов и т.п.)

Документация (мануалы, гайды, FAQ, helps и т.п.)

Форматы даты и времени

Формат чисел

Формат денежных величин

Поддержка различных календарей

Изображения (картинки, иконки)

Звук (в частности, озвучка, если таковая имеется)

Реклама (текстовая, аудио, видео)

И т.д.

#### **Глобализация (G11N)**

Завершая обзорную часть, хотелось бы ещё упомянуть термин [Globalization](https://en.wikipedia.org/wiki/Internationalization_and_localization) (G11N), который используется для обозначения комбинации I18N и L10N. Также он используется для обозначения процесса, благодаря которому компания может выпустить свой продукт на глобальном рынке. Честно говоря не скажу, что есть конкретное определение глобализации вашего продукта (т.е. использования такого количества локалей, после которого ваша глобализация достигает 100%), поэтому я бы охарактеризовал процесс глобализации формулой вида: G11N = I18N + n\*L10N, где n — количество локалей, используемой вашим продуктом.

Напоследок я бы хотел ещё указать сокращение для Translation — [T9N](https://www.adverbum.com/en/blog/globalization-internationalization-localization-translation). На практике используется редко, но тоже встречается.

**SQL** databases are relational, **NoSQL** are non-relational. **SQL** databases use structured query language and have a predefined schema. **NoSQL** databases have dynamic schemas for unstructured data. ... **SQL** databases are table based, while **NoSQL** databases are document, key-value, graph or wide-column stores

The Secure Shell Protocol is a cryptographic network protocol for operating network services securely over an unsecured network. Typical applications include remote command-line, login, and remote command execution, but any network service can be secured with SSH.

**Отличие** от **библиотеки**

«**Фреймворк**» отличается от понятия **библиотеки** тем, что **библиотека** может быть использована в программном продукте просто как набор подпрограмм близкой функциональности, не влияя на архитектуру программного продукта и не накладывая на неё никаких ограничений.

Главное **отличие класса от интерфейса** — в том, что **класс** состоит из **интерфейса** и реализации. Любой **класс** всегда неявно объявляет свой **интерфейс** — то, что доступно при использовании **класса** извне. ... Кроме этого **интерфейса**, у **класса** есть также реализация — методы Вставить, Повернуть, Вынуть и их вызов в методе Открыть.

## **Главное**

* Инкапсулируйте все, что может изменяться;
* Уделяйте больше внимания интерфейсам, а не их реализациям;
* Каждый класс в вашем приложении должен иметь только одно назначение;
* Классы — это их поведение и функциональность.

## **Базовые принципы ООП**

* Абстракция — отделение концепции от ее экземпляра;
* Полиморфизм — реализация задач одной и той же идеи разными способами;
* Наследование — способность объекта или класса базироваться на другом объекте или классе. Это главный механизм для повторного использования кода. Наследственное отношение классов четко определяет их иерархию;
* Инкапсуляция — размещение одного объекта или класса внутри другого для разграничения доступа к ним.

## **Используйте следующее вместе с наследованием**

* Делегация — перепоручение задачи от внешнего объекта внутреннему;
* Композиция — включение объектом-контейнером объекта-содержимого и управление его поведением; последний не может существовать вне первого;
* Агрегация — включение объектом-контейнером ссылки на объект-содержимое; при уничтожении первого последний продолжает существование.

**Risk**-**based testing** (RBT) is a type of software **testing** that functions as an organizational principle used to prioritize the **tests** of features and functions in software, **based** on the **risk** of failure, the function of their importance and likelihood or impact of failure

**Some risk based testing techniques are given below:**

* Product **Risk Management** (PRisMa)
* Pragmatic **Risk Analysis** and **Management** (PRAM)
* Systematic Software **Testing** (SST

## Как защитить сайт от SQL-инъекции

* 1. Белые списки
  2. Не пользуйтесь методом GET в формах
  3. Проверять откуда данные
  4. PDO
  5. Используйте функции trim (убирает лишние пробелы), htmlspecialchars (заменяет треугольные скобки и другие спецсимволы), addslashes (экранирует кавычки и специальные символы) и другие

В чем разница между VNC и VPN? … VPN использует более продвинутые технологии, например, протоколы туннелирования. Это способ создать частную сеть в более крупной общественной сети, например, в интернете. VNC позволяет вам управлять удаленными компьютерами, обычно через VPN.

XML – это язык разметки (как это фактически говорится в его названии), тогда как JSON – это способ представления объектов (как также отмечено в его названии

SOAP — это формат протокола, основанный на XML, тогда как REST — это архитектурный подход.

Модель **OSI** является общей, независимой от протокола, но большинство протоколов и систем придерживаются ее, в то время как модель **TCP**/**IP** основана на стандартных протоколах, которые разработал интернет.

**Метод POST** подразумевает, что Вы передаёте данные в теле запроса. **Метод PUT** подразумевает, что Вы передаёте всё, что нужно в URL. Тела запроса нет.

SSL — Secure Socket Layer, уровень защищенных сокетов. TLS — Transport Layer Security, безопасность транспортного уровня. SSL является более ранней системой, TLS появился позднее и он основан на спецификации SSL 3.0, разработанной компанией Netscape Communications.

**DNS** и **DHCP** работают на архитектуре клиент-сервер, но имеют разные термины. **DNS** сопоставляет имя домена с IP-адресом, тогда как **DHCP** - это протокол, который назначает IP-адреса хостам в сети, будь то статически или динамически. **DHCP** также используется при настройке **DNS**-сервера на хост

Непрерывная интеграция (CI), непрерывная доставка (CD) и непрерывное развёртывание (CD) — devops-подход к разработке и апгрейду программного обеспечения, подразумевающий непрерывное, конвейерное тестирование, сборку, доставку и развёртывание обновлений. Возможно как отдельное применение компонентов этого подхода (CI или CI + CD), так и их последовательное использование в рамках единого процесса (CI + CD + CD).

## **Непрерывная доставка (CD)**

Непрерывная доставка (CD) — CI + CD. Следующий после CI уровень. Теперь новая версия не только создаётся и тестируется при каждом изменении кода, регистрируемом в репозитории, но и может быть оперативно запущена по одному нажатию кнопки развёртывания. Однако запуск развёртывания всё ещё происходит вручную — ту самую кнопку всё же надо кому-то нажать. Этот метод позволяет выпускать изменения небольшими партиями, которые легко изменить или устранить в случае необходимости.

## **Непрерывное развёртывание (CD)**

Непрерывное развёртывание (CD) — CI +CD + CD. После автоматизации релиза остаётся один ручной этап: одобрение и запуск развёртывания в продакшен (злосчастная кнопка). Практика непрерывного развёртывания упраздняет и это, не требуя непосредственного утверждения со стороны разработчика. Все изменения развёртываются автоматически.

SQL Injection - подмена или вставка sql запроса в базу данных

XSS-атака - замена кода в сайте с целью обмана пользователя,подмена сайта целиком

Code Injection - Inject code in system/program

Fiddler - Traffic packages manager

Wireshark - TRAFFIC Manager

Настройка Android Studio - Запуск проекта, интерфейс на основе intellij, есть декомпиляция apk файлов.

Тестирование UI - тестирование интерфейса пользователя - тестирование функциональности приложения со стороны пользователя

Автоматизация тестов с помощью фреймворка Espresso - Тестирование ui через Android Studio

Автоматизация с использованием Appium - эмуляция сервера для android приложений во время разработки

Javascript - высокоуровневый язык для веб инфраструктуры, Исполняет запросы пользователя на клиенсткой машине

Отладка, Developer Tools - коммандная панель браузера которая дает разработчику возможности мененджмента страницы

Типы данных в JavaScript

В JavaScript есть 8 основных типов.

number для любых чисел: целочисленных или чисел с плавающей точкой; целочисленные значения ограничены диапазоном ±(253-1).

bigint для целых чисел произвольной длины.

string для строк. Строка может содержать ноль или больше символов, нет отдельного символьного типа.

boolean для true/false.

null для неизвестных значений – отдельный тип, имеющий одно значение null.

undefined для неприсвоенных значений – отдельный тип, имеющий одно значение undefined.

object для более сложных структур данных.

symbol для уникальных идентификаторов.

Оператор typeof позволяет нам увидеть, какой тип данных сохранён в переменной.

Имеет две формы: typeof x или typeof(x).

Возвращает строку с именем типа. Например, "string".

Для null возвращается "object" – это ошибка в языке, на самом деле это не объект.

(npm) - developer package manager with command line

Асинхронность - Это прием программирования, который использует многпоточность, выполняя несколько операций одновременно, во избежание блокировки.

Платформа Node.js - платформа расширяющая возможности javascript в серверах и в железе

Jest - testing network for Javascript projects (requers nvm/node js)

Puppeeter - testing framework for Chromium

Playwright - Testing framork for Chrome,Firefox and Safari

Cypress - End to End Javascrypt Testing tool