

1. Почему практики тест-дизайна нельзя применять сразу после получения требований?

Сначала нужно проанализировать требования и разбить их до атомарного вида, выявить серые зоны, создать mindmap и блок-схему.

2. В какой ситуации классы эквивалентности и граничные значения могут существовать по отдельности? Аргументируй свой ответ и приведи примеры.

Если класс — это набор, прописывают все значения, которые в него входят. У таких классов границ не бывает. Например, список праздников.

3. Что такое эквивалентность? Что такое класс эквивалентности?

КЭ - диапазон или набор значений, на которые система реагирует одинаково.

4. Можно ли исключить проверку в середине диапазона в пользу проверок на границах, входящих в диапазон? Аргументируй свой ответ.

Нельзя, нужно проверить как работает система при всех возможных вариантах.

5. Представь, что тебе нужно протестировать форму: у каждого поля есть валидатор. Результат работы формы зависит от комбинации данных в полях. Какие практики тест-дизайна следует применить и почему? Аргументируй свой ответ.

КЭ и ГЭ - для максимального покрытия вводимых значений.
Позитивные и негативные проверки - для тестирования валидных и невалидных значений.
Попарное тестирование - для оптимизации проверок.

6. Какими способами можно оптимизировать количество проверок при работе с таблицами принятия решений? Аргументируй свой ответ.

Использовать технику попарного тестирования. Она поможет сократить количество возможных комбинаций до тех пор, когда необходимые параметры будут присутствовать хотя бы в одной комбинации.

7. Опиши, чем чек-лист отличается от тест-кейсов. Приведи примеры, где применяют и то, и другое.

Чек-лист - это список проверок, которые нужно выполнить. Чек-лист не содержит шагов воспроизведения.
Тест-кейс - это проверка одного конкретного элемента с шагами воспроизведения, ID, ОР, ФР.
Логике расчета стоимости такси можно проверить тест-кейсом, а интерфейс заказа такси чек-листом.

8. Как правильно составить баг-репорт? Какие элементы баг-репорта — обязательные? Почему?

ID - уникальный идентификатор бага. ID у багов не может повторяться
Заголовок - Что? Где? Когда? Заголовок коротко и понятно описывает суть проблемы
Шаги воспроизведения - детальный список шагов, которые помогут воспроизвести баг
ФР - то, что получилось
ОР - то, что должно было получиться
Приоритет - отображает важность бага и срочность его исправления
Окружение - указывается техника, на которой был найден баг ОС, ПО и их версии

Баг-репорт составляется при обнаружении ошибки. Баг - это не соответствие фактического результата от ожидаемого.
Для того чтобы баг-репорт был составлен правильно, необходимо знать что он в себя включает и как должны быть сформулированы и выглядеть каждый из его пунктов.
Баг-репорты должны содержать подробности, позволяющие читателю понять природу бага. То есть в них должно быть достаточно сведений, чтобы понять, почему описываемое поведение — баг, и как его воспроизвести. Вместе с тем нужно стараться не включать в баг-репорт лишнего и не повторяться.

9. По каким правилам составляют заголовок баг-репорта? Что будет, если составить заголовок неправильно?

Заголовок должен отвечать на вопросы Что? Где? Когда? Правильно описывать суть проблемы, чтобы проверяющий его понял и уделит достаточно внимания на исправление
Если составить заголовок неправильно, то разработчик может не понять важность бага и не взяться за него

10. Из чего состоит клиент-серверная архитектура приложения? Кратко опиши функциональность каждого элемента.

Клиент - локальный компьютер, который выполняет отправку запроса к серверу
Сервер - система обрабатывающая запросы от клиента и формирующая необходимую информацию в ответ
Клиент предоставляет пользовательский графический интерфейс, формулирует и отправляет запросы к серверу, получает итоги запросов
Сервер: хранит и защищает данные, работает с поступающими запросами от клиента, отправляет ответ клиенту

Интернет - связующее звено, с помощью которого происходит взаимодействие сервера и клиента.

11. Опиши этапы обработки запроса после того, как в адресную строку браузера вводят URL: <https://yandex.ru>.

Браузер обращается к DNS-серверу
Браузер соединяется с сервером
Браузер отправляет HTTP-запрос на сайт, для получения контента
Сервер обрабатывает запрос
Сервер отправляет браузеру ответ
Браузер обрабатывает полученные данные, отображает сайт

Основное предназначение DNS-серверов — хранение информации о доменах и ее предоставление по запросу пользователей, а также кэширование DNS-записей других серверов.

12. Что такое кэш? Зачем он нужен? Какое правило нужно соблюдать при работе с кэшем в тестировании?

Кэш - данные веб-страницы, которые сохраняются на компьютер, когда мы открываем изображения, слушаем аудиозаписи или смотрим видео
При тестировании необходимо очищать кэш, т.к. из-за того, что данные закешированы может происходить баг

Кэш нужен для того, чтобы при повторном обращении к той или иной странице она загрузилась быстрее - браузер заберет файлы из кэш-хранилища, а не с сервера.

13. Ты знаешь, что есть протоколы HTTP и HTTPS.

- Чем отличаются HTTP и HTTPS? В каких случаях не стоит пользоваться HTTP?
- Из каких компонентов состоит HTTP запрос: за что отвечает каждый?
- Какие HTTP-методы ты знаешь? За что они отвечают? Приведи примеры применения разных методов.
- Что такое код HTTP-ответа? Какие коды бывают?

HTTPS - протокол передачи информации с шифрованием.
Не стоит использовать HTTP в тех случаях, если будет использоваться передача каких-то конфиденциальных данных
Стартовая строка. Описывает метод и путь до ресурса
Заголовок запроса - информация от клиента к серверу
Тело сообщения - информация, передаваемая от клиента к серверу
GET - получение информации (информация о товарах в корзине)
POST - создание и отправка данных (передача формы)
DELETE - удаление данных (удаление продуктов)
PUT - изменение данных (добавление продуктов в корзину)
Код HTTP-ответа - трехзначное число, показывающее код результата
1xx - информационное сообщение
2xx - сообщение об успешном действии
3xx - перенаправление
4xx - ошибка в запросе клиента
5xx - ошибка со стороны сервера

Тело сообщения может отсутствовать, но стартовая строка и заголовок являются обязательными элементами.

14. Опиши, из каких компонентов может состоять URL. За что отвечает каждый из них?

Протокол передачи данных HTTP или HTTPS
Логин:пароль - для понимания сервером, какой пользователь к нему обращается
Имя хоста:порт - доменное имя или IP адрес, на который идет запрос
Порт - виртуальный порт, имеющий номер, через который идет подключение
Путь - расположение ресурса или какого либо файла в файловой системе сервера
Параметры запроса - дополнительная информация в виде "ключ=значение" в файловой системе сайта
Фрагмент текста, "якорь" - помогает попасть на нужную часть веб-страницы

15. Какие виды мобильных приложений бывают? В чём особенность каждого?

Нативные - необходима установка на мобильное устройство. Более производительны, т.к. используют ресурсы устройства.
Веб-приложения - используют клиент-серверную архитектуру. По сути являются сайтами, адаптированными под работу с мобильного устройства.
Гибридные - комбинация двух вышеописанных видов. Требуе установки приложения и постоянных доступ к интернету

16. Чем эмулятор отличается от реального устройства? Кратко опиши недостатки и преимущества при тестировании.

Эмулятор - приложение, которое позволяет создать виртуальное устройство. Позволяет имитировать работу различных мобильных устройств, с различными параметрами.
Преимущества: позволяет покрывать множество мобильных устройств при тестировании
Недостатки: ресурсоемкость, требует производительного ПК. Нет возможности проверить устройство на нагрев, разрядка аккумулятора, связь, различные виды тапов и др.

17. Проверь, есть ли ошибки в JSON

Если да, то приложи верный вариант файла.

```
01 { "name": {
02   "id": "1",
03   "value": "Файл",
04   "list":
05     "items": {
06       "new_doc": {
07         "value": "Новый",
08         "onclick": "create_new_doc",
09         "open_doc": {
10           "value": "Открыть...",
11           "onclick": "open_doc",
12           "save_doc": {
13             "value": "Сохранить",
14             "onclick": "save_doc",
15             "save_as_doc"
16             "сохранить как...",
17             "onclick": "save_as_doc",
18             "print_option": {
19               "value": "Параметры печати",
20               "onclick":
21                 "show_print_option": {
22                   "text": "Напечатать",
23                   "color": "Белая печать?", " ",
24                   "Размер печати": "A4"}}
25             }
26           }
27         }
28       }
```

Ссылка на иллюстрацию:

<https://code.s3.yandex.net/ga/schemes/diploma-29.png>

```
{  "Меню": {
    "id": "1",
    "value": "Файл",
    "list": {
      "items": {
        "new_doc": {
          "value": "Новый",
          "onclick": "create_new_doc"
        },
        "open_doc": {
          "value": "Открыть...",
          "onclick": "open_doc"
        },
        "save_doc": {
          "value": "Сохранить",
          "onclick": "save_doc"
        },
        "save_as_doc": {
          "value": "Сохранить как...",
          "onclick": "save_as_doc"
        },
        "print_option": {
          "value": "Параметры печати",
          "onclick": {
            "show_print_option": {
              "Цвет": "Насыщенный",
              "Черно—белая печать?": "",
              "Размер печати": "A4"
            }
          }
        }
      }
    }
  }
}
```

18. Что такое реляционная база данных? Чем она отличается от нереляционной?

Реляционная - это база данных, которая состоит из таблиц и связей между ними
Реляционная база данных отличается от нерялиционной тем, что в первой информация хранится в таблицах, а во второй данные хранятся в формате ключ:значение без использования таблиц

19. Напиши, какие виды JOIN бывают. В чем особенность каждого?

INNER JOIN - внутреннее соединение. Этот вид джойна выведет только те строки, если условие соединения выполняется (является истинным, т.е. TRUE)
LEFT JOIN - возвращаются все строки левой таблицы. В случае, если данных нет, возвращается значение NULL
RIGHT JOIN - возвращаются все строки правой таблицы. В случае, если данных нет, возвращается значение NULL
FULL JOIN - возвращаются строки и левой, и правой таблиц. Если у строк левой таблицы и правой таблицы выполняется условие соединения, они объединяются в одну строку. В случае, если данных нет, возвращается значение NULL.

Исходные данные для заданий ниже

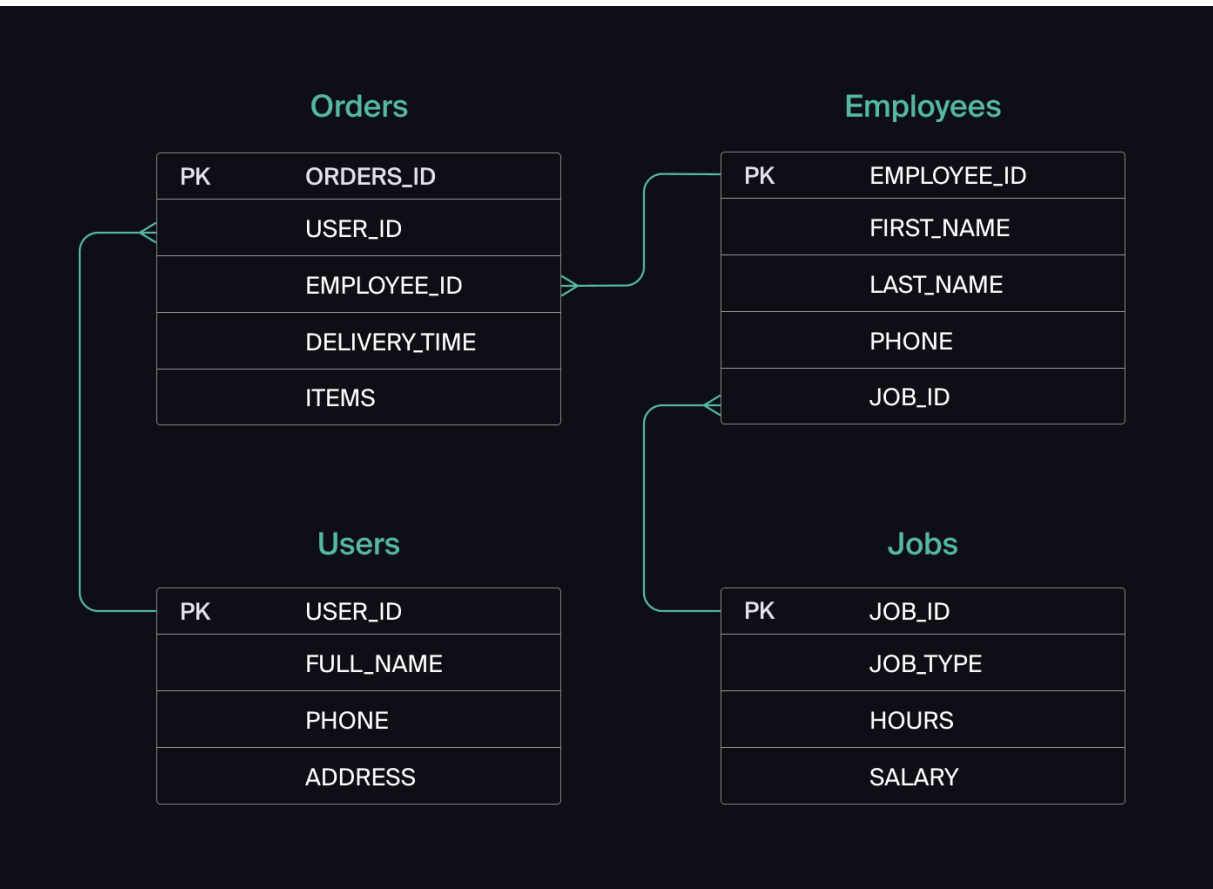
Ты тестируешь сервис, который доставляет еду за 30 минут. Пока это маленький стартап, поэтому ты работаешь всего с четырьмя таблицами:

Orders — все доставленные заказы;
ORDERS_ID — ID заказов, int;
USER_ID — ID пользователей, int;
EMPLOYEE_ID — ID сотрудников, int;
DELIVERY_TIME — время доставки в минутах, int;
ITEMS — список товаров, char;

Users — пользователи;
USER_ID — ID пользователей, int;
FULL_NAME — полное ФИО пользователя, char;
PHONE — номер телефона пользователя, char;
ADDRESS — адрес пользователя, char;

Employees — работники;
EMPLOYEE_ID — ID сотрудника, int;
FIRST_NAME — имя сотрудника, char;
LAST_NAME — фамилия сотрудника, char;
PHONE — телефон сотрудника, char;
JOB_ID — ID специализации, int;

Jobs — типы работ в сервисе
JOB_ID — ID специализации, int;
JOB_TYPE — тип специализации, char;
HOURS — число рабочих часов в неделю, int;
SALARY — зарплата сотрудника с данной специализацией в рублях, int;



Ссылка на иллюстрацию:

<https://code.s3.yandex.net/ga/schemes/diploma-33.png>

20. В службу поддержки пришло много жалоб: заказы, в которых есть гречка, доставляют почти час, хотя сервис обещает успеть в 30 минут.

Проверь, действительно ли курьеры опаздывают. Выбери все заказы, где есть хотя бы один товар - «гречка» и время доставки свыше 30 минут. В результирующей таблице должны быть ID заказов и ID курьеров.

В ответе приложи SQL-запрос.

SELECT ORDERS_ID, EMPLOYEE_ID FROM Orders WHERE ITEMS LIKE "%гречка%" AND DELIVERY_TIME>30;

21. Менеджер предложил добавить новую функциональность в продукт: мониторинг, который показывает самых активных клиентов за всё время работы компании.

Проверь, что список пользователей корректно выводится на экран. На этом этапе разработки достаточно проверить только ID клиентов.

Выбери пять самых активных клиентов по количеству заказов.

В результирующую таблицу выведи ID каждого пользователя и число заказов.

Отсортируй данные по убыванию числа заказов, выбери пять самых активных клиентов.

В ответе приложи SQL-запрос.

SELECT USER_ID, COUNT(ORDERS_ID) AS ORDERS_CNT FROM Orders GROUP BY USER_ID ORDER BY ORDERS_CNT DESC LIMIT 5

22. Из бухгалтерии пришёл баг-репорт: зарплаты сотрудников рассчитываются некорректно. Оказалось, что почти все ошибки в расчётах — в расчётных листах менеджеров.

Выведи список ID всех сотрудников, у которых в специализации содержится «менеджер», с зарплатой больше 70 000 рублей.

В ответе приложи SQL-запрос.

SELECT EMPLOYEE_ID
FROM Employees
INNER JOIN Jobs ON Employees.JOB_ID = Jobs.JOB_ID
WHERE Jobs.JOB_TYPE like '%менеджер%'
AND Jobs.SALARY > 70000;

23. Изучи три ситуации и ответь на вопрос: стоит ли писать автотесты в этом случае? Аргументируй свой ответ.

- 1) Проект существует давно, у него написано много ручных тестов.
- 2) Проект временный: продлится всего несколько месяцев.
- 3) Проект нестабилен: в функциональность часто вносят изменения.

1. Необходимо написать автотесты для экономии времени и не тестировать все вручную
2. Не стоит, процесс написания автотестов трудоемкий и может занять много времени
3. Лучше применить ручное тестирование, т.к. код может часто меняться