

Вопрос/Ответ		Комментарий
1. Почему практики тест-дизайна нельзя применять сразу после получения требований?		
<p>Потому что сначала нужно провести тест-анализ для уточнения серых зон и декомпозиции требований.</p> <p>Таким образом, это делается для оптимизации трудозатрат (чтобы не пришлось переделывать тестовую документацию).</p>		
2. В какой ситуации классы эквивалентности и граничные значения могут существовать по отдельности? Аргументируй свой ответ и приведи примеры.		
<p>В ситуации, когда выделенный класс является набором, а не диапазоном значений.</p> <p>В таком случае нельзя однозначно выделить границы.</p> <p>Примеры:</p> <ul style="list-style-type: none">Строка состоящая из русских букв. Это набор (если только не указаны параметры сортировки этого набора).Строка содержащая спец. символы.Пустое поле.		
3. Что такое эквивалентность? Что такое класс эквивалентности?		
<p>Эквивалентность - равнозначность/равноценность.</p> <p>КЭ - класс, в пределах которого ожидается одинаковое поведение тестируемой системы/приложения/сервиса.</p>		
4. Можно ли исключить проверку в середине диапазона в пользу проверок на границах, входящих в диапазон? Аргументируй		

Вопрос/Ответ		Комментарий
<p>На мой взгляд - лучше не исключать.</p> <p>Причина - зачастую, при программировании, на границах диапазонов появляются ошибки в следствии невнимательности при итерации в циклах.</p> <p>Соответственно, учитывая эту погрешность, лучше не пренебрегать проверкой в середине диапазона, т.к. она исключает вероятность этой погрешности.</p> <p>Вдобавок, если баг внутри диапазона, то и весь диапазон скорее всего будет воспроизводить этот баг, т.к. это один КЭ.</p>		
<p>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.</p>		
<p>Стоит применить:</p> <p>1) КЭ и ГЗ (по валидности значений)</p> <p>2а) Таблицу принятия решений (если комбинаций немного)</p> <p>2б) Попарное тестирование (если комбинаций много)</p> <p>КЭ и ГЗ используется в качестве базового разбиения (это золотой стандарт), таблица принятия решений - для визуализации проверок, а попарное тестирование стоит применять, если нужно оптимизировать количество проверок ещё больше (уникальные комбинации).</p>		
<p>6. Какими способами можно оптимизировать количество проверок при работе с таблицами принятия решений? Аргументируй</p>		
<p>Можно оптимизировать попарным тестированием до уникальных комбинаций.</p> <p>Таким образом, можно значительно сократить кол-во проверок, при этом сохранив область покрытия тестирования.</p>		
<p>7. Опиши, чем чек-лист отличается от тест-кейсов. Приведи примеры, где применяют и то, и другое.</p>		

Вопрос/Ответ		Комментарий
<p>Чек-лист менее детализирован. Он содержит список элементов, которые нужно протестировать.</p> <p>В тест-кейсе большая детализация. В ТК присутствует пошаговое описание и проверяется только один элемент.</p> <p>ЧЛ и ТК применяются, в зависимости от договорённостей внутри команды.</p> <p>Если в команде есть стажёры/джуны, которые не настолько хорошо знакомы с системой, то для тестирования ими используют тест-кейсы.</p> <p>Как правило, опытные сотрудники чаще работают по ЧЛ.</p> <p>Также, если система не притерпевает постоянных изменений большого кол-ва функционала, то использование ТК более оправдано, в том числе для создания автотестов на их основе.</p>		
8. Как правильно составить баг-репорт? Какие элементы баг-репорта — обязательные? Почему?		
<p>Баг репорт нужно составлять с подробным и понятным описанием, т.е. так, чтобы ошибку можно было легко воспроизвести.</p> <p>Обязательные элементы:</p> <p>1) ID - для придания уникальности и простоты отслеживания цикла жизни бага.</p> <p>2) Заголовок - для описания проблемы.</p> <p>3) Шаги воспроизведения - для упрощения воспроизведения бага.</p> <p>4) ОР и ФР - для сравнения и понимания, что это баг.</p> <p>5) Окружение - для воспроизведения и локализации.</p> <p>6) Приоритет - для оптимизации выстраивания очереди на исправления багов.</p>		
9. По каким правилам составляют заголовок баг-репорта? Что будет, если составить заголовок неправильно?		
<p>Заголовок должен быть исчерпывающим и понятным.</p> <p>По нему должно быть ясно в чём суть бага.</p> <p>Должен отвечать на вопросы: Что? Где? Когда?</p> <p>Если составить заголовок неправильно, то будет непонятно в чём суть бага, когда и где он возникает.</p>		
10. Из чего состоит клиент-серверная архитектура приложения? Кратко опиши функциональность каждого элемента.		

Вопрос/Ответ		Комментарий
<p>Состоит из:</p> <p>1) Клиент (пользовательская машина/интерфейс, например - браузер пользователя)</p> <p>2) Сервер (удалённая машина, на которой происходит обработка пользовательских запросов и с которой формируются ответы на эти запросы)</p> <p>3) База данных (удалённая машина, на которой хранится информация сервиса/приложения)</p> <p>4) Сеть интернет (по ней происходит передача пакетов данных между звеньями архитектуры)</p>		
11. Опиши этапы обработки запроса после того, как в адресную строку браузера вводят URL: https://yandex.ru.		
<p>Этапы:</p> <p>1) Инициализация клиентом запроса к DNS</p> <p>2) DNS возвращает клиенту IP-адрес</p> <p>3) Клиент формирует запрос по IP-адресу к серверу</p> <p>4) Сервер обрабатывает клиентский запрос</p> <p>5) Сервер возвращает клиенту данные страницы и код ответа</p> <p>6) Клиент принимает данные и код</p> <p>7) Клиент отрисовывает полученные данные в виде страницы "yandex.ru"</p>		
12. Что такое кэш? Зачем он нужен? Какое правило нужно соблюдать при работе с кэшем в тестировании?		
<p>Кэш - данные вэб страниц, которые хранятся локально на клиенте.</p> <p>Кэш данные нужны для оптимизации количества запросов к серверу и для хранения информации о странице.</p> <p>При тестировании, стоит соблюдать правило очистки кэша для того, чтобы происходило обновление данных посредством обмена новыми данными с сервером.</p> <p>В противном случае, данные не обновятся и при тестировании можно увидеть старые баги, которые вроде как уже исправили.</p>		
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?		

Вопрос/Ответ		Комментарий
<p>HTTP отличается от HTTPS отсутствием шифрования. HTTP не стоит пользоваться при работе с чувствительными данными (банковские операции, авторизация/регистрация, паспортные данные итд)</p> <p>HTTP состоит из:</p> <ul style="list-style-type: none">1) Стартовой строки (метод, путь, версия протокола)2) Заголовков запроса (информация о клиенте, параметры запроса, характеристики контента)3) Тела запроса (данные с фронта) <p>HTTP методы:</p> <ul style="list-style-type: none">1) POST (отправка данных на сервер, например - при отправке данных формы регистрации)2) GET (запрос данных с сервера, например - переход по ссылке на ресурс/сервис)3) PUT (изменение данных на сервере, например - изменение данных пользователя в личном кабинете)4) DELETE (удаление данных на сервере, например - удаление аккаунта пользователя) <p>Код HTTP-ответа - это кодовое обозначение результата запроса. Коды можно условно разделить на следующие подгруппы:</p> <ul style="list-style-type: none">1) Информационные сообщения "1xx"2) Сообщения об успехе "2xx"3) Редирект(перенаправление) "3xx"4) Сообщения об ошибке клиента "4xx"5) Сообщения об ошибке сервера "5xx"		
14. Опиши, из каких компонентов может состоять URL. За что отвечает каждый из них?		
<p>Компоненты:</p> <ul style="list-style-type: none">1) Схема (протоколы запросов)2) Логин (идентификация пользователя)3) Пароль (идентификация пользователя)4) Имя хоста/порт (домен, IP к которому обращается клиент)5) Путь (путь к ресурсу, или его конкретной части)6) Параметры запроса (доп. параметры ключ=значение)7) Якорь (доп. параметр # выступающий как указатель нужной части страницы)		
15. Какие виды мобильных приложений бывают? В чём особенность каждого?		

Вопрос/Ответ		Комментарий
<p>Виды моб. приложений:</p> <p>1) Вэб приложения (доступ через браузер, находятся на сервере)</p> <p>2) Нативные приложения (устанавливаются локально на устройство)</p> <p>3) Гибридные (гибрид вэб и нативных, основа устанавливается локально, контент подгружается через сеть с сервера)</p>		
16. Чем эмулятор отличается от реального устройства? Кратко опиши недостатки и преимущества при тестировании.		
<p>Главное отличие в том, что эмулятор - программа, а не реальное устройство.</p> <p>Эмулятор не может передать весь функционал реального устройства, а только лишь часть его функций.</p> <p>Вдобавок, эмулятор не взаимодействует с тем же набором технических компонентов что и реальное устройство.</p> <p>Преимущества:</p> <p>1) Быстрота развёртывания различных эмулируемых устройств</p> <p>2) Цена тестирования (не надо покупать устройства)</p> <p>Недостатки:</p> <p>1) Неполноценность в покрытии тестовыми сценариями, которые задействуют расширенный функционал устройства.</p> <p>2) Нет взаимодействия с тем же набором технических компонентов что и на реальном устройстве.</p> <p>3) Симулируемый девайс не повторит с точностью поведение оригинала</p> <p>4) Настройка Android studio :)</p>		
17. Проверь, есть ли ошибки в JSON Если да, то приложи верный вариант файла.		

Вопрос/Ответ			Комментарий
<pre>01 { "Меню": { 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 "Цвет": "Насыщенный", 23 "Черно-белая печать?": "", 24 "Размер печати": "A4"}} 25 } 26 } 27 } 28 }</pre>			
Ссылка на иллюстрацию:			
https://code.s3.yandex.net/qa/schemes/diploma-29.png			

Вопрос/Ответ		Комментарий
<pre>{ "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"}} } } } } }</pre>		
18. Что такое реляционная база данных? Чем она отличается от нереляционной?		
<p>Реляционные БД - БД, состоящая из табличного представления данных и связей между этими таблицами.</p> <p>Отличие от нереляционных в том, что нереляционные используют представление в виде объектов по типу ключ-значение (например в виде JSON)</p>		
19. Напиши, какие виды JOIN бывают. В чем особенность каждого?		

Вопрос/Ответ		Комментарий
<p>Виды JOIN-ов:</p> <p>1)INNER JOIN - формирует выборку только из данных, у которых выполнено условие присоединения.</p> <p>2)OUTER JOIN (LEFT - возвращаются все строки левой таблицы с дополнениями по совпадению условиям из правой таблицы, RIGHT - возвращаются все строки правой таблицы с дополнениями по совпадению условиям из левой таблицы, FULL - возвращаются строки и левой, и правой таблиц в виде одной строки по совпадению условиям.)</p>		
<p>Исходные данные для заданий ниже</p> <p>Ты тестируешь сервис, который доставляет еду за 30 минут. Пока это маленький стартап, поэтому ты работаешь всего с четырьмя таблицами:</p> <p>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;</p> <p>Users — пользователи; USER_ID — ID пользователей, int; FULL_NAME — полное ФИО пользователя, char; PHONE — номер телефона пользователя, char; ADDRESS — адрес пользователя, char;</p> <p>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;</p> <p>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;</p>		

Вопрос/Ответ			Комментарий																																				
<div><div><div>Orders</div><table><tr><td>PK</td><td>ORDERS_ID</td></tr><tr><td></td><td>USER_ID</td></tr><tr><td></td><td>EMPLOYEE_ID</td></tr><tr><td></td><td>DELIVERY_TIME</td></tr><tr><td></td><td>ITEMS</td></tr></table></div><div><div>Employees</div><table><tr><td>PK</td><td>EMPLOYEE_ID</td></tr><tr><td></td><td>FIRST_NAME</td></tr><tr><td></td><td>LAST_NAME</td></tr><tr><td></td><td>PHONE</td></tr><tr><td></td><td>JOB_ID</td></tr></table></div><div><div>Users</div><table><tr><td>PK</td><td>USER_ID</td></tr><tr><td></td><td>FULL_NAME</td></tr><tr><td></td><td>PHONE</td></tr><tr><td></td><td>ADDRESS</td></tr></table></div><div><div>Jobs</div><table><tr><td>PK</td><td>JOB_ID</td></tr><tr><td></td><td>JOB_TYPE</td></tr><tr><td></td><td>HOURS</td></tr><tr><td></td><td>SALARY</td></tr></table></div></div>		PK	ORDERS_ID		USER_ID		EMPLOYEE_ID		DELIVERY_TIME		ITEMS	PK	EMPLOYEE_ID		FIRST_NAME		LAST_NAME		PHONE		JOB_ID	PK	USER_ID		FULL_NAME		PHONE		ADDRESS	PK	JOB_ID		JOB_TYPE		HOURS		SALARY		
PK	ORDERS_ID																																						
	USER_ID																																						
	EMPLOYEE_ID																																						
	DELIVERY_TIME																																						
	ITEMS																																						
PK	EMPLOYEE_ID																																						
	FIRST_NAME																																						
	LAST_NAME																																						
	PHONE																																						
	JOB_ID																																						
PK	USER_ID																																						
	FULL_NAME																																						
	PHONE																																						
	ADDRESS																																						
PK	JOB_ID																																						
	JOB_TYPE																																						
	HOURS																																						
	SALARY																																						
Ссылка на иллюстрацию:																																							
https://code.s3.yandex.net/qa/schemes/diploma-33.png																																							
<p>20. В службу поддержки пришло много жалоб: заказы, в которых есть гречка, доставляют почти час, хотя сервис обещает успеть в 30 минут.</p> <p>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.</p> <p>В ответе приложи SQL-запрос.</p>																																							

Вопрос/Ответ		Комментарий
SELECT ORDERS_ID, EMPLOYEE_ID FROM Orders WHERE ITEMS LIKE '%гречка%' AND DELIVERY_TIME > 30;		
<p>21. Менеджер предложил добавить новую функциональность в продукт: мониторинг, который показывает самых активных клиентов за всё время работы компании.</p> <p>Проверь, что список пользователей корректно выводится на экран. На этом этапе разработки достаточно проверить только ID клиентов.</p> <p>Выбери пять самых активных клиентов по количеству заказов.</p> <p>В результирующую таблицу выведи ID каждого пользователя и число заказов.</p> <p>Отсортируй данные по убыванию числа заказов, выбери пять самых активных клиентов.</p> <p>В ответе приложи SQL-запрос.</p>		
SELECT USER_ID, COUNT(*) AS ORDERS_COUNT FROM Orders GROUP BY USER_ID ORDER BY COUNT(*) DESC LIMIT 5;		
<p>22. Из бухгалтерии пришёл баг-репорт: зарплаты сотрудников рассчитываются некорректно. Оказалось, что почти все ошибки в расчётах — в расчётных листах менеджеров.</p> <p>Выведи список ID всех сотрудников, у которых в специализации содержится «менеджер», с зарплатой больше 70 000 рублей.</p> <p>В ответе приложи SQL-запрос.</p>		

Вопрос/Ответ		Комментарий
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) Проект существует давно , у него написано много ручных тестов. - однозначно стоит , т.к. требования полностью сформированы и вероятность необходимости переписывать автотесты стремится к нулю. Вдобавок, основа в виде ручных тестов уже подготовлена. 2) Проект временный : продлится всего несколько месяцев. - 50/50 , т.к. в зависимости от сроков, использование автотестов может не окупиться . Можно автоматизировать только основной функционал , если позволят сроки. 3) Проект нестабилен : в функциональность часто вносят изменения. - нет смысла , т.к. требования плавающие и придётся постоянно корректировать автотесты на лету, что невыгодно и нереально воплощать в жизнь.		