# Письменные ответы на вопросы

**Вопрос 1**

Какую технику тест-дизайна ты используешь, чтобы составить тест-кейсы для задачи ниже? Обоснуй свой выбор.

Задача

На сайте застройщика нужно реализовать следующую функциональность: поиск квартиры для покупки по определенным параметрам. Все фильтры — с одиночным выбором. Варианты такие:

Название ЖК: «Золотое сияние», «Небесный полет».

Тип квартиры: студия, 1-комнатная, 2-комнатная, 3-комнатная, 4-комнатная.

Этаж: 1, 2, 3, 4, 5, 6, 7, 8, 9.

Отделка: черновая, чистовая.

Балкон: есть/нет.

**Ответ 1**

можно использовать технику тест-дизайна — таблицу принятия решений.

Её используют, когда в требованиях есть разные комбинации условий. Их удобно представить графически, в виде таблицы, чтобы точно покрыть тестами все требования и ничего не упустить,будет 360 тест кейсов, но это долго,

цена ошибки не слишком высока (Так бывает при тестировании медицинского ПО или ПО для самолётов) и поэтому не очень подходит.

лучше использовать технику попарного тестирования

Когда параметров и их значений много, попарное тестирование сокращает количество проверок в несколько раз

она проверит только основные комбинации где будут задействованы разные праметры таблицы

например:

1. Название ЖК:

-«Золотое сияние»

- «Небесный полет».

2. Тип квартиры:

- Студия

- 1-комнатная

- 2-комнатная

- 3-комнатная

- 4-комнатная

3. Этаж:

- 1

- 2

- 3

- 4

- 5

- 6

- 7

- 8

- 9

4. Отделка:

- Черновая

- Чистовая

5. Балкон:

- Есть

- Нет

Далее, для каждой группы входных данных мы можем выбрать параметр и составить тест-кейсы, учитывая все возможные комбинации значений параметров:

- Тест-кейс 1: «Золотое сияние», Студия, 2, черновая, есть балкон.

- Тест-кейс 2: «Золотое сияние», 1-комнатная, 5, черновая, нет балкона.

- Тест-кейс 3: «Небесный полет», 2-комнатная, 9, чистовая, есть балкон.

- Тест-кейс 4: «Небесный полет»,3-комнатная, 3, чистовая, нет балкона.

- Тест-кейс 5: «Золотое сияние», 4-комнатная, 1, черновая, есть балкон.

**Вопрос 2**

Посмотри на тест-кейс ниже. Что стоит изменить и почему?

Заголовок:

Отображение главной странице Яндекс Практикума для неавторизованного пользователя

Предусловие:

Пользователь не авторизован в системе Яндекс Практикума

Шаги воспроизведения:

Открой главную страницу Яндекс Практикума https://practicum.yandex.ru/

Ожидаемый результат:

Открылась главная страница для неавторизованного пользователя

**Ответ 2**

В этом тест кейсе не понятно что именно мы проверяем, расхождение в заголовке с О.Р.

Написано "Отображение главной страницы", т.е мы проверяем отображение? Если так,

что должно отобразиться? Лучше свериться с требованиями, макетами и конкретно указать, что мы ожидаем увидеть, что за главная страница, как она должна выглядеть. Исходя из этого сформировать О.Р.

Если мы проверяем открывается ли главная страница, как указано в О.Р, то следует написать заголовок: Открывается главная страница яндекс практикума для неавторизованного пользователя.

-Уникальный и полный заголовок

не повторяет заголовки других тест-кейсов — чтобы не запутаться;

конкретный и отвечает на вопрос «Что я проверяю?» или «Что? Где? Когда?»

В этом тест-кейсе отсутствует обязательный параметр:

-Уникальный ID по которому мы можем ссылаться на этот документ или например прикрепить его к баг-репорту.

ID тест-кейса не должен повторять другие. Если будет два одинаковых, команда запутается.

Иногда ID для тест-кейсов задаёт система, в которой работает тестировщик. Если такого нет, следуют правилам, принятым в команде. О них тебе расскажут коллеги.

ID может содержать цифры, текст или и то и другое. Например, YNDX-2.

-Ожидаемый результат должен быть однозначным

Однозначный результат — тот, который одинаково представит любой человек из команды. Такой результат отвечает на вопросы «Что именно?», «Где?», «Когда?».

Рекомендация:

Добавить ID. окружение, приложить ссылку на макеты и требования, новичку будет проще разобраться.

**Вопрос 3**

Ты тестируешь чек-бокс «Заплатить сразу» в форме создания заказа. Ожидаемый результат — чек-бокс отмечен, в базе данных это поле содержит true. Но на самом деле в базе почему-то стоит false. Это ошибка — её нужно локализовать. Тебе известно, что архитектура приложения включает фронтенд, бэкенд и базу данных. Опиши, что нужно сделать в Devtools, чтобы локализовать ошибку.

**Ответ 3**

1.Для локализации ошибки,

необходимо воспроизвести проблему на фронтенде — через UI

-открыть Devtools, кликнуть правой кнопкой мыши на веб-странице и выбрав "посмотреть код" или нажать F12 на клавиатуре, сочетание клавиш: для Windows — Ctrl+Shift+i

Включить параметр Disable cache. Иначе все запросы будут сохраняться в памяти компьютера

Devtools, вкладка "Elements"(Элементы) позволяет просмотреть HTML-код веб-страницы и изменить значения элементов. В данном случае, нужно убедиться, что чек-бокс "Заплатить сразу" имеет правильные атрибуты, такие как name и value, и что он отмечен (checked).

-в UI выполнить создание заказа, перейти в Devtools, вкладка "Network"(Сеть)

-проверить какие HTTP-запросы отправляются на бэкенд

какие данные принимаются в ответ. Здесь следует обратить внимание на название и данные запроса, связанные с созданием заказа, а также на код ответа и данные, полученные от бэкенда. Возможно, есть какие-то ошибки в передаче данных между фронтендом и бэкендом, которые могут привести к неправильному значению поля в базе данных.

-Devtools, вкладка "Console" можно просмотреть сообщения об ошибках JavaScript или выводы, которые могут быть полезными для локализации причины ошибки. Здесь могут быть видны возможные ошибки в логике фронтенда, которые могут приводить к неправильной обработке данных и неправильной передаче их на бэкенд.

2. Воспроизвести проблему на бэкенде — через API с помощью Postman

Если ошибка не связана с фронтендом, необходимо перейти к анализу бэкенда. Для этого необходимо изучить соответствующий код серверной части, связанный с созданием заказа, и проверить, какие данные принимает и обрабатывает сервер, а также какие данные записывает в базу данных. Возможно, есть ошибки в логике серверной части, которые приводят к неправильной обработке данных и сохранению неверных значений в базе данных.

3. В SQL-Postgre проверяем базы данных . Проверить значение поля чек-бокса "Заплатить сразу" в базе данных и сравнить его с ожидаемым результатом.

Если причина ошибки не была найдена на данном этапе, может потребоваться дальнейшее диагностирование, такое как анализ логов приложения или обращение за помощью к другим разработчикам или администраторам системы. Возможно, ошибку может вызывать что-то более сложное и специфичное.

**Вопрос 4**

Возможна ли такая ситуация: отправленный GET-запрос удаляет всю информацию в базе данных? Обоснуй свой ответ.

**Ответ 4**

Теоретически, возможно, что отправленный GET-запрос может удалить всю информацию в базе данных, однако это необычная ситуация и не соответствует стандартным методам и протоколам HTTP.

Стандартно, GET-запросы предназначены для получения данных из сервера, а не для их изменения или удаления. Они являются безопасными и идемпотентными, то есть многократное выполнение одного и того же GET-запроса не должно изменить состояние сервера.

Однако, некорректная или небезопасная реализация сервера может обработать GET-запрос таким образом, что он приведет к удалению информации в базе данных. Например, если GET-запрос не использует защищенный ресурс, такой как REST API, и сервер некорректно обрабатывает параметры запроса, то это может привести к удалению данных.

Такая ситуация может быть результатом программной ошибки, недостаточной проверки входных данных на стороне сервера или вредоносного поведения, такого как SQL-инъекция. Однако, добросовестные разработчики и администраторы баз данных обычно применяют меры безопасности и соответствуют принятому подходу к разработке веб-приложений, чтобы избежать подобных проблем.

**Вопрос 5**

В базе данных есть две таблицы. Какие поля стоит сделать первичными ключами, а какие — внешними? Обоснуй своё решение.

Таблица сотрудников employee с полями:

id — идентификатор сотрудника,

fio — ФИО сотрудника,

position id — идентификатор должности.

Таблица должностей position с полями:

id — идентификатор должности,

name — название должности,

salary — зарплата на данной должности.

**Ответ 5**

Первичный ключ может быть не только номером, но и набором любых символов. Главное, чтобы ключ не повторялся.

id — идентификатор сотрудника, стоит сделать первичным ключом Первичный ключ сделает запись уникальной и позволит избежать полных дубликатов в таблице.

id — идентификатор должности, так же стоит сделать первичным ключом

position id — идентификатор должности, стоит сделать внешним связующим ключом к таблице position

**Вопрос 6**

Напиши запрос, который выведет данные в таком формате: ФИО сотрудника, название должности и зарплата.

**Ответ 6**

select e.fio, p.name, p.salary

from position as p

JOIN employee as e on p.id = e.position\_id

**Вопрос 7**

Код ниже должен проверять, что оператор // действительно выполняет деление на целое число. Чего здесь не хватает и почему?

def test\_integer\_division():

a = 5//2;

assert a == 2

**Ответ 7**

assert a == 2

удобнее сделать эту проверку автоматической. Здесь пригодится ключевое слово assert (англ. «утверждать») его нужно подставить перед оператором сравнения ==. Оно работает как логическое выражение: возвращает True или False. Например, если сумма чисел правда 2, вернёт True, а иначе — False

Если проверка не прошла, программа вернёт сообщение об ошибке — AssertionError

**Вопрос 8**

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

**Ответ 8**

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

Например:

-список праздников.

-выбор величины, Нм, см, дм, м...

-вид животного: котика, собачку, попугайчика, тараканчика и тд.

-размер футболки: S, XS, M, L, XL, XXL, XXXL, XXXXXXXXXXL. больше размеров не бывает.

**Вопрос 9**

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

**Ответ 9**

мы исходим из гипотезы, что "на границах класса эквивалентности вероятность бага больше" - т.е. что поведение на границах может отличаться от поведения в середине диапазона.

Мы выполняем дополнительные однотипные тесты (в сумме - пять значений из одного класса) для того, чтобы побольше симулировать работу пользователя. Чтобы внести элемент случайности в работу с системой. Так мы повышаем вероятность случайно найти неожиданный баг.

Если внутренние проверки не являются важными для логики алгоритма или не представляют существенного интереса с точки зрения выполнения операций, их исключение может упростить чтение и понимание кода.

Однако, есть несколько случаев, когда исключение проверок в середине диапазона может быть нежелательным или даже опасным.

--Первый случай — это когда проверки в середине диапазона необходимы для правильного функционирования алгоритма или для обеспечения безопасности данных. Например, если внутри диапазона выполняются некие изменения данных или операции, важные для корректного выполнения программы, и пропустив их, можно нарушить логику или породить ошибки в программе.

--Второй случай — это когда необходимо обрабатывать все значения внутри диапазона, даже если они не граничные. Некоторые алгоритмы могут требовать полной обработки всех возможных значений в диапазоне, и пропуск проверок в середине диапазона может привести к некорректным результатам или неполной обработке данных.

чтобы определить, можно ли исключить проверки в середине диапазона

-необходимо внимательно изучить логику и цели проверок внутри диапазона, а также понять, как это может повлиять на работу алгоритма или программы вцелом.

В некоторых случаях этот подход может быть эффективным,

но в других случаях его использование может привести к некорректным результатам или ошибкам.

Поэтому решение должно быть основано на конкретных требованиях и особенностях задачи.

**Вопрос 10**

Ты тестируешь мобильную версию Яндекс Аренды. Там обнаружился баг: при переходе в раздел «Показать на карте» появляется сообщение «Непредвиденная ошибка», и приложение закрывается. Разработчик просит воспроизвести баг и прислать логи. Как это сделать?

**Ответ 10**

Логи можно смотреть через Dev Tools во вкладке Console

Нажми правой кнопкой на пустое место на текущей веб-странице → «Исследовать элемент» → вкладка Console.

Теперь в левом верхнем углу кликни на кнопку Show console sidebar — это прямоугольник с треугольником внутри.

Здесь есть классификация логов по степени важности.

Errors — ошибки, которые влияют на взаимодействие пользователя с приложением. Например, из-за таких ошибок может не работать часть приложения.

Warnings — предупреждения о потенциальных ошибках, при этом приложение всё равно смогло выполнить запрос.

Information — информационные сообщения. Например, о работе базы данных.

# Отчёт о тестировании

Нахожусь В процессе доработки чек листа

## Функциональное тестирование веб-приложения

Приложение проверено на стенде (<https://0d63fa71-8b19-41c8-b511-0e8989f4fb1a.serverhub.praktikum-services.ru/order>).

При составлении проверок применялись была составлена Maind map <https://xmind.app/m/K8nkMX>

Все известные требования были покрыты чек-листом: [Елена Ёлшина когорта 8а -QA+](https://docs.google.com/spreadsheets/d/1WWiTBzFT0p5zs-tpkC3urxsnS0_OEJ7dk-34mPw7-hc/edit?usp=sharing)).

Результаты выполнения тестов можно посмотреть здесь: [Елена Ёлшина когорта 8а -QA+](https://docs.google.com/spreadsheets/d/1WWiTBzFT0p5zs-tpkC3urxsnS0_OEJ7dk-34mPw7-hc/edit?usp=sharing)). Из (количество тестов) успешно прошло (количество успешных тестов), не прошло — (количество не пройденных тестов).

Список багов, найденных при тестировании, разбит по приоритетам:

1. Блокирующие:  
- [ссылки в виде списка]

2. Критичные:  
- [[https://stronghel.youtrack.cloud/issue/MEEK86S-23/Chrome.Ne-sozdaetsya-zakaz-v-UI-pri-nazhatii-na-knopku-zakazat](https://stronghel.youtrack.cloud/issue/MEEK86S-23/Chrome.Ne-sozdaetsya-zakaz-v-UI-pri-nazhatii-na-knopku-zakazat)

3. Средний приоритет:  
-

4. Низкий приоритет:  
- [ссылки в виде списка]

Заключение:  
\*ответь на несколько вопросов ниже в свободной форме

1. Какой баг показался самым критичным?
2. На твой взгляд, какая самая «хитрая» серая зона есть в требованиях?
3. Проверенная тобой функциональность готова к релизу? Почему?

## Ретест багов в мобильном приложении

Был проверен фикс багов. Из них не исправлено (1), исправлено — (3).

Список багов можно посмотреть здесь: [[https://stronghel.youtrack.cloud/issue/TTFOTYSWA-2/Pri-klike-na-notifikaciyu-prilozhenie-lomaetsya]](https://stronghel.youtrack.cloud/issue/TTFOTYSWA-2/Pri-klike-na-notifikaciyu-prilozhenie-lomaetsya%5D)

. <https://stronghel.youtrack.cloud/issue/TTFOTYSWA-3/Vo-vkladke-Moi-otobrazhayutsya-ne-tolko-moi-zakazy-no-i-vse-prinyatye-zakazy>

<https://stronghel.youtrack.cloud/issue/TTFOTYSWA-4/Pri-filtracii-po-stancii-metro-ostayutsya-vse-zakazy-a-ne-tolko-dlya-vybrannoj-stancii>

<https://stronghel.youtrack.cloud/issue/TTFOTYSWA-5/Ne-poyavlyaetsya-skroll-v-bloke-filtracii-po-stanciyam-esli-dobavit-bolshe-vosmi-stancij>

## Регрессионное тестирование мобильного приложения по готовым тест-кейсам

Результаты выполнения регрессионных тестов можно посмотреть здесь: ([Елена Ёлшина когорта 8а -QA+](https://docs.google.com/spreadsheets/d/1WWiTBzFT0p5zs-tpkC3urxsnS0_OEJ7dk-34mPw7-hc/edit?usp=sharing)).

Из (10) успешно прошло (0), не прошло — (10).

Список багов, найденных при тестировании, разбит по приоритетам:

1. Блокирующие:  
- [ссылки в виде списка]

2. Критичные:  
- <https://stronghel.youtrack.cloud/issue/MEEK86S-22/posle-nazhatiya-vtorym-kurerom-knopki-prinyat-prilozhenie-lomaetsya>

3. Средний приоритет:  
- <https://stronghel.youtrack.cloud/issue/MEEK86S-21/Ne-vozmozhno-prinyat-zakaz-iz-predusloviya-zakaz-propadaet-iz-spiska-vse-ekrana-kurera.>

4. Низкий приоритет:  
- [ссылки в виде списка]

Заключение:  
\*ответь на несколько вопросов ниже в свободной форме

1. Какой баг показался самым критичным?
2. Такой продукт можно выпускать в релиз? Почему?

Ссылка на гитхаб <https://github.com/stronghel/C-Users-PycharmProjects-Elena_Elshina_Kagorta_8a_QA-_diplom>

Выводы о проделанной работе  
Проведя смоук тестирование обнаружила критический баг в веб-версия сайта Яндекс самокат в браузере гугл хром

Также в Мобильной версии по истечению небольшого времени приложение ломается самопроизвольно.

Отправляю работу на проверку, того что сделано, нахожусь на стадии доработки чек листа продолжаю тестирование.

Как для тебя прошла первая практическая часть проекта? С какими сложностями пришлось столкнуться? Что получилось хорошо, а что не очень? Какие мысли остались?