# 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.

Здесь удобнее всего применить технику тест-дизайна - попарное тестирование, так как в требованиях много параметров. При возможности воспользуюсь сервисом для составления таблиц попарного тестирования. В данном случае таблица будет состоять из:

* Название ЖК:
* - корректное значение: «Золотое сияние»,
* - корректное значение: «Небесный полет»,
* - некорректное значение: пустое значение,
* - некорректное значение: название другого ЖК.
* Тип квартиры:
* - корректное значение: студия,
* - корректное значение: 1-комнатная,
* - корректное значение: 2-комнатная,
* - корректное значение: 3-комнатная,
* - корректное значение: 4-комнатная,
* - некорректное значение: пустое значение,
* - некорректное значение: неверный тип квартиры.
* Этаж: (# с помощью классов эквивалентности и граничных значений выбираем для проверки корректных значений три варианта)
* - корректное значение: 1,
* - корректное значение: 5,
* - корректное значение: 9,
* - некорректное значение: пустое значение,
* - некорректное значение: неверное значение этажа
* Отделка:
* - корректное значение: черновая,
* - корректное значение: чистовая,
* - некорректное значение: пустое значение,
* - некорректное значение: неверный тип отделки
* Балкон:
* - корректное значение: есть
* - корректное значение: нет,
* - некорректное значение: пустое значение,
* - некорректное значение: неверный тип балкона.

Такая таблица поможет визуализировать уникальные комбинации параметров и рассмотреть каждый из них при составлении тест-кейсов.

Однако, следует отметить, что один из принципов тестирования звучит так: избыточное тестирование невозможно. А статистика говорит о том, что вероятность ошибки уменьшается, если проверить больше комбинаций. Так что при наличие ресурсов, времени и вдохновения, для охвата всех возможных комбинаций, можно воспользоваться таблицей принятия решений, она помогает ничего не пропустить, но в данном случае тестирование по ней займет много времени.

## **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

Обязательными элементами тест-кейса являются: ID, заголовок, шаги, ожидаемый результат (ОР). Также можно указывать: предусловие/постусловие, операционную систему (ОС), устройство, на котором тестируют продукт, другие компоненты: например, версию ПО, названия элементов, которые проверяют, роль пользователя.

В представленном тест-кейсе отсутвует ID. Заголовок тест-кейса во-первых содержит грамматическую ошибку (главной страницЫ), во-вторых должен быть более информативным. Шаги воспроизведения содержат приказной тон (повелительное наклонение). Ожидаемый результат крайне расплывчат, по нему невозможно определить что именно мы хотим получить, как определить, что открываемая страница именно для неавторизованного пользователя, т.е. нужно более конкретно описать, что пользователь увидит на главной странице, если он не авторизован на сайте.

Также, при условии, если операционные системы, браузеры, расширения не указаны отдельно, т.е. если это не обобщенный тест-кейс для разных устройств, расширений, ос, то в предусловиях или шагах воспроизведения необходимо указывать где именно происходит проверка.

**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.

Для локализации данной ошибки, следует использовать Devtools, чтобы проанализировать взаимодействие между фронтендом, бэкендом и базой данных.

1. Открываем Devtools (при открытой форме создания заказа);
2. Переходим на вкладку “Сеть”, где мы видим все запросы отправляемые браузером и ответы сервера на эти запросы;
3. Ищем запрос, который отправляется при нажатии на чек-бокс “Заплатить сразу”;
4. В данном запросе проверяем отправленные данные. Если в отправленных данных значение чек-бокса “Заплатить сразу” не равно true, то скорее всего проблема на стороне фронтенда;
5. Если в отправленных данных значение чек-бокса верное, переходим на вкладку “Ответ” для просмотра ответа сервера;
6. В ответе с сервера проверяем значение чек-бокса, если ответ возвращается со значением false то скорее всего проблема на стороне бэкенда или базы данных;
7. Если в ответе сервера значение чек-бокса равно true, то вероятнее всего, что ошибка на стороне базы данных.

В данном случае требуется исследовать какие операции с базой данных выполняются при этом сценарии.

## **Вопрос 4**

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

Такая ситуация не должна быть возможна, потому что метод GET является идемпотентным, запрашивает данные и не предназначен для удаления. Основное предназначение данного метода - получение данных, а не их изменение или удаление. Т.е. при стандартной работе, GET- запрос не должен иметь незапланированных результатов в виде удаления информации в базе данных.

Однако, в зависимости от языка реализации и прав доступа к базе данных, вы можете делать все, что вам нравится, в запросе GET. Извлечение в GET и обновление с помощью POST-запросов - это просто соглашение которое остается на усмотрение разработчика. Но все же для чистоты создания продукта, безопасности его использования не стоит пренебрегать устоявшимися правилами.

## **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.

В таблице базы данных не должно быть дубликатов, т.е все записи (строки) в таблице должны быть уникальны. Для этого к каждой записи добавляется уникальный признак — первичный ключ, такой ключ не позволит создать абсолютно одинаковые записи. Внешний ключ — это значение которое применяется для связи между данными в двух таблицах с целью контроля данных, иными словами это поле (или набор полей) в одной таблице, которое ссылается на первичный ключ в другой таблице.

В первой таблице первичным ключом будет id - идентификатор сотрудника, так как это значение уникально для каждого сотрудника. Остальные значения в полях могут повторяться.

Во второй таблице первичным ключом будет id - идентификатор должности, так как в этой таблице оно уникально.

Внешним ключом связывающим эти две таблицы будет position\_id - идентификатор должности, которое является первичным ключом в таблице должностей и является частью таблицы сотрудников.

## **Вопрос 6**

Посмотри информацию из пятого вопроса и напиши запрос, который выведет данные в таком формате: ФИО сотрудника, название должности и зарплата.

SELECT e.fio,

p.name,

p.salary

FROM employee AS e

INNER JOIN position AS p ON e.position\_id = p.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;

a == 2

test\_integer\_division();

В данной функции не хватает assert перед a == 2

Assert - ключевое слово,оно работает как логическое выражение: возвращает True или False. В данном случае оно необходимо для проверки правильности работы функции.

Основная цель указанной функции - сравнивать результат выполнения оператора // с ожидаемым значением и выводить ошибку если они не совпадают. При использовании ключевого слова assert, функция будет останавливаться если значение а не равно 2. Сейчас же программа присваивает результат деления переменной, а не проверяет его правильность.

## **Вопрос 8**

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

Класс эквивалентности (КЭ) — это набор входных или выходных данных ПО, которые обрабатываются программой по одному алгоритму или приводят к одному результату.

Граничные значения (ГЗ) — это первая и последняя точки класса эквивалентности.

Исходя из этих определений граничные значения не могут существовать отдельно от классов эквивалентности. Техника выделения ГЗ помогает проверить, корректно ли приложение обрабатывает границы КЭ, а также дополнить проверки КЭ типа «диапазон» тестами на границах.

Однако КЭ могут существовать отдельно от ГЗ. У Классов эквивалентности типа набор (совокупность значений, объединенных одним свойством) не будет граничных значений. Объекты внутри набора нельзя упорядочить или сравнить между собой. Необходимо качественное, а не количественное сравнение значений набора с условием требования. Например: в интернет-магазине домашних питомцев в меню мы выбираем вид животного: кошечку, собачку, хомячка, тараканчика, …

Однако следует отметить, что иногда проверка граничного значения в классе эквивалентности может также являться проверкой другого КЭ. Например: для поля “Имя” ограничение от 5 до 20 символов, так для класса эквивалентности “менее 5 символов”, граничное значение 0 символов, может быть и проверкой класса эквивалентности “пустое поле”.

## **Вопрос 9**

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

Считаю, что исключать проверку в середине диапазона нельзя.

1. Один из принципов тестирования звучит так: избыточное тестирование невозможно.

2. Предполагается, что при проверке определенного диапазона, все значения будут обработаны приложением одинаково, и на выходе дадут одинаковый результат. Однако на границах чаще всего возникают ошибки, необходимость заложена из предпосылок, что при написании кода, разработчик может ошибиться при указании границ и/или логики. Поведение продукта на границах и в середине диапазона может отличаться, так что правильность поведения на границах не гарантирует правильность поведения в середине диапазона.

## **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?

Исходя из текста вопроса, можно сделать вывод, что баг обнаружился при проведении тестирования мобильной версии приложения по ранее составленному чек-листу или тест-кейсам. Т.е. с разработчиком уже ранее было оговорено в каком виде он желает видеть обнаруженные ошибки и вероятнее всего это составленные баг-репорты.

Так что в данном случае, я считаю, что необходим составить баг-репорт который будет содержать:

1. ID;
2. Название: “ При переходе (# желательно указать откуда переходим) в раздел “Показать на карте” появляется сообщение “Непредвиденная ошибка””;
3. Предусловие (# указание нужного тестового стенда, либо ссылка на тестируемую версию приложения);
4. Шаги воспроизведения (# подробные и понятные для воспроизведения данного бага);
5. Ожидаемый результат (# согласно требований или логики при отсутствии оных);
6. Фактический результат: “Появляется сообщение “Непредвиденная ошибка””;
7. Окружение на котором производится тестирование;
8. Приложить скринкаст воспроизведения ошибки;
9. Приложить файл с логами (# например воспользоваться специальной программой для записи логов, выбранной в зависимости от устройства. Например для Android, можно воспользоваться Android Debug Bridge (ADB), чтобы получить логи приложения. Для этого необходимо будет подключить устройство к компьютеру с установленной программой. В командной строке ввести: adb logcat - d > logs.txt и полученный файл пригладываем);
10. Отправляем баг-репорт разработчику.

# Отчет о тестировании

## Функциональное тестирование веб-приложения

Приложение проверено на стенде [(ссылка на стенд)](https://ba0ef696-e6d5-45ca-8a1e-27630f52cfb5.serverhub.praktikum-services.ru/). Все известные требования были покрыты чек-листом: ([ссылка на google sheets с чек-листом](https://docs.google.com/spreadsheets/d/19_KS83R2iDqsY9PHFieLZiJKiKNrYUwEykHNC7IO4rM/edit?usp=sharing))

Результаты выполнения тестов можно посмотреть здесь: ([ссылка на google sheets с чек-листом](https://docs.google.com/spreadsheets/d/19_KS83R2iDqsY9PHFieLZiJKiKNrYUwEykHNC7IO4rM/edit?usp=sharing)). Из 142 тестов успешно прошло 122 теста, не прошло — 20 тестов.

Список багов, найденных при тестировании, разбит по приоритетам:

1. Блокирующие:  
- [EK12VD-18](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-18) Отсутствует переход на следующую страницу при клике на кнопку "Да" на всплывающем окне "Хотите оформить заказ"

2. Критичные:  
- [EK12VD-14](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-14) Возможный выбор сегодняшней и прошедшей даты в выпадающем календаре в поле "Когда привезти самокат" в форме "Про аренду"

3. Средний приоритет:  
- [EK12VD-3](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-3) Отключается страница при вводе неправильного названия в поле "Станция метро"

- [EK12VD-4](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-4) Поле "Имя" в форме "Для кого самокат" не принимает ввод данных с тире

- [EK12VD-6](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-6) Поле "Фамилия" в форме "Для кого самокат" принимает ввод данных более 15 символов

- [EK12VD-7](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-7) Неверная логика приема полем "Адрес" в форме "Для кого самокат" разрешенных данных

- [EK12VD-10](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-10) Поле "Адрес" в форме "Для кого самокат" не принимает ввод 50 символов.

- [EK12VD-11](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-11) Поле "Телефон" в форме "Для кого самокат" не принимает ввод 10 символов.

- [EK12VD-12](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-12) Поле "Телефон" в форме "Для кого самокат" принимает ввод 13 символов.

- [EK12VD-13](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-13) Кнопка Далее активна при не заполненном поле "Адрес" и заполненных остальных полях

- [EK12VD-15](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-15) Возможный ввод данных вручную в поле "Когда привезти самокат" в форме "Про аренду"

- [EK12VD-16](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-16) Не появляется окно с номером заказа при нажатии на кнопку "Заказать" в форме "Про аренду"

- [EK12VD-17](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-17) Размытый текст названий станций метро в выпадающем списке в поле "Станция метро" в форме "Для кого самокат"

4. Низкий приоритет:  
- [EK12VD-1](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-1) Несоответствие названия поля "Адрес: куда привезти самокат" в форме "Для кого самокат"

- [EK12VD-9](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-9) Отсутствие сообщения об ошибке в поле "Адрес" в форме "Для кого самокат" при вводе 0 символов.

- [EK12VD-2](https://ekaterinakuranina.youtrack.cloud/issue/EK12VD-2) Несанкционированная заливка в полях формы "Для кого самокат" при автозаполнении

Заключение:

1. Какой баг показался самым критичным?

Самым критичным багом является невозможность сделать заказ.

1. На твой взгляд, какая самая «хитрая» серая зона есть в требованиях?

Понятие “хитрая” очень растяжимое, но серых зон много. Вот некоторые из них, которые на мой взгляд достаточно “хитрые”:

Ограничения для поля “Адрес” в форме “Для кого самокат” прописаны таким образом, что введение просто улицы уже считается корректным вводом. Да и в принципе было бы неплохо привязывать адрес к актуальной карте.

В поле “Комментарий для курьера” в форме “Про аренду” не предусмотрен вывод ошибки при нарушении ограничений.

1. Проверенная тобой функциональность готова к релизу? Почему?

В данном случае релиз невозможен, потому что есть блокирующий баг. Который сводит на нет всю идею данного продукта. Заказ сделать невозможно.

## Ретест багов в мобильном приложении

Был проверен фикс багов. Из них не исправлено - 1 баг, исправлено — 3 бага, обнаружено новых - 1 баг

Список багов можно посмотреть здесь: (добавлено окружение)

1. [retest\_samokat-7](https://ekaterinakuranina.youtrack.cloud/issue/retest_samokat-7/Vo-vkladke-Moi-otobrazhayutsya-ne-tolko-moi-zakazy-no-i-vse-prinyatye-zakazy) Во вкладке «Мои» отображаются не только мои заказы, но и все принятые заказы
2. [retest\_samokat-6](https://ekaterinakuranina.youtrack.cloud/issue/retest_samokat-6/Pri-klike-na-notifikaciyu-prilozhenie-lomaetsya) При клике на нотификацию приложение ломается
3. [retest\_samokat-8](https://ekaterinakuranina.youtrack.cloud/issue/retest_samokat-8/Pri-filtracii-po-stancii-metro-ostayutsya-vse-zakazy-a-ne-tolko-dlya-vybrannoj-stancii) При фильтрации по станции метро остаются все заказы, а не только для выбранной станции
4. [retest\_samokat-9](https://ekaterinakuranina.youtrack.cloud/issue/retest_samokat-9/Ne-poyavlyaetsya-skroll-v-bloke-filtracii-po-stanciyam-esli-dobavit-bolshe-vosmi-stancij) Не появляется скролл в блоке фильтрации по станциям, если добавить больше восьми станций

## Регрессионное тестирование мобильного приложения по готовым тест-кейсам

Результаты выполнения регрессионных тестов можно посмотреть здесь: [(ссылка на google sheets с тестами)](https://docs.google.com/spreadsheets/d/1FWXj49QMZoAusyHMqV0XLPyoNM6GogdGQo6SuZG3XXU/edit?usp=sharing).

Из 10 тестов успешно прошло 2 теста, не прошло — 8 тестов из которых 1 пропущен, по причине невозможности воспроизведения.

Список багов, найденных при тестировании, разбит по приоритетам:

1. Блокирующие:  
- [retest\_samokat-12](https://ekaterinakuranina.youtrack.cloud/issue/retest_samokat-12) Отсутствие сообщения об ошибки принятия заказа, если заказ отменился.

- [retest\_samokat-13](https://ekaterinakuranina.youtrack.cloud/issue/retest_samokat-13) Отсутствие сообщения об ошибки принятия заказа, если заказ взял другой курьер

2. Критичные:  
- [retest\_samokat-16](https://ekaterinakuranina.youtrack.cloud/issue/retest_samokat-16) Не отображается комментарий в полной карточке заказа

.

3. Средний приоритет:  
- [retest\_samokat-10](https://ekaterinakuranina.youtrack.cloud/issue/retest_samokat-10) Дублируется карточка заказа во вкладке "Мои" после нажатия на кнопку "Принять" в короткой карточке

-[retest\_samokat-11](https://ekaterinakuranina.youtrack.cloud/issue/retest_samokat-11) Дублируется карточка заказа во вкладке "Мои" после нажатия на кнопку "принять" в Полной карточке

[retest\_samokat-15](https://ekaterinakuranina.youtrack.cloud/issue/retest_samokat-15) У вкладки “Мои” синяя точка до принятия заказа

[retest\_samokat-17](https://ekaterinakuranina.youtrack.cloud/issue/retest_samokat-17) Не отображается информация о цвете в полной карточке заказа, при незаполненном поле Цвет

4. Низкий приоритет:  
- [retest\_samokat-14](https://ekaterinakuranina.youtrack.cloud/issue/retest_samokat-14) После принятия карточка заказа уезжает из списка “Все” — с анимацией движения влево

Заключение:

1. Какой баг показался самым критичным?

Самыми критичными в этом случае, на мой взгляд, являются блокирующие баги, когда приложение совсем перестает работать

1. Такой продукт можно выпускать в релиз? Почему?

В данной версии присутствуют блокирующие баги, при которых приложение полностью перестает функционировать и его нужно перезапускать. При этих багах работа курьера фактически стопорится. Считаю, что в таком виде выпускать нельзя, нужно фиксить.

Выводы о проделанной работе

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

Первая часть прошла неплохо. Особых сложностей не возникло. На вопрос, что получилось хорошо, а что не очень ответит ревьюер, мне кажется, что все получилось неплохо. Задания понравились, особенно то, что были затронуты вопросы всей теоретической части.