

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.

Я бы использовала метод попарного тестирования для составления тест-кейсов для данной задачи. Он заключается в том, что все возможные пары входных данных тестируются в отдельности. Этот метод обеспечит максимальное покрытие функциональности при относительно небольшом количестве тест-кейсов. В данном случае, у нас есть несколько параметров с различными вариантами выбора, и метод попарного тестирования позволит нам проверить все возможные комбинации пар входных данных, что обеспечит достаточное покрытие функциональности.

Например, если у нас есть параметры - "Название ЖК", "Этаж", "отделка" и "Балкон" для "Название ЖК" - данные «Золотое сияние», «Небесный полет», и т.д

то метод попарного тестирования позволит нам покрыть все возможные комбинации пар входных данных, такие как

«Золотое сияние»,	студия	1	черновая	есть
-------------------	--------	---	----------	------

и т.д.

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

Нужно добавить описание ожидаемых элементов на странице и исправить орфографию, чтобы тест-кейс был более конкретным и понятным. Например:

Заголовок:

Отображение главной страницы Яндекс Практикума для неавторизованного пользователя

Предусловие:

Пользователь не авторизован в системе Яндекс Практикума

Шаги воспроизведения:

1. Перейти по ссылке <https://practicum.yandex.ru/>

2. Проверить, что на странице отображается логотип Яндекс Практикума

Ожидаемый результат:

Открылась главная страница Яндекс Практикума для неавторизованного пользователя.

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 ошибки необходимо выполнить следующие шаги в Devtools:

1. Открыть консоль разработчика и проверить, что при выборе чек-бокса "Заплатить сразу" отправляется запрос на сервер.
2. Проверить, что в запросе отправляется значение true для поля "Заплатить сразу".
3. Проверить, что на сервере значение true сохраняется в базе данных.
4. Если на сервере значение сохраняется как false, то проследить, где происходит изменение значения поля "Заплатить сразу" и почему оно не соответствует ожидаемому результату.
5. Использовать инструменты для отладки бэкенда и базы данных, чтобы определить, где происходит изменение значения поля и как это исправить.

Вопрос 4

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

Нет, такая ситуация не возможна. GET-запросы используются только для получения информации из базы данных, они не могут изменять или удалять данные. Для изменения или удаления данных необходимо использовать POST, PUT или DELETE запросы. Если в процессе выполнения GET-запроса происходит удаление данных, то это может быть связано с ошибкой в коде бэкенда или базы данных.

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 "employee" поле "id" должно быть установлено как первичный ключ, поскольку оно представляет собой уникальный идентификатор каждого сотрудника. Поле "position_id" должно быть установлено как внешний ключ, поскольку оно связывает таблицу "employee" с таблицей "position" по значению идентификатора должности.

В таблице "position" поле "id" должно быть установлено как первичный ключ, поскольку оно представляет собой уникальный идентификатор каждой должности. Ни одно поле из таблицы "employee" не связано с значениями в таблице "position", поэтому не требуется устанавливать какое-либо поле внешним ключом.

Вопрос 6

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

```
SELECT employee.fio, position.name, position.salary
FROM employee
JOIN position ON employee.position_id = position.position_id;
```

Вопрос 7

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

Скопировать кодJSX

```
def test_integer_division():  
    a = 5//2;  
    a == 2
```

```
test_integer_division();
```

В коде не хватает проверки на равенство ожидаемого результата и фактического. Необходимо добавить `assert a == 2`, чтобы убедиться, что оператор `//` выполняет деление на целое число и возвращает ожидаемый результат. Итоговый код будет выглядеть так:

```
def test_integer_division():
```

```
    a = 5//2
```

```
    assert a == 2
```

```
test_integer_division();
```

Вопрос 8

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

Классы эквивалентности и граничные значения могут существовать по отдельности в случае, когда граничные значения не являются представителями какого-либо класса эквивалентности. Например, при тестировании функции, которая принимает на вход возраст пользователя и возвращает соответствующее сообщение, можно выделить классы эквивалентности по диапазонам возрастов (например, до 18 лет, от 18 до 60 лет, старше 60 лет). Однако, при этом также нужно учитывать граничные значения, например, возраст 18 лет, который является граничным значением между первым и вторым классами эквивалентности.

Таким образом, классы эквивалентности и граничные значения могут существовать по отдельности в тех случаях, когда граничные значения не принадлежат ни одному из классов эквивалентности.

Вопрос 9

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

В некоторых случаях можно исключить проверку в середине диапазона в пользу проверок на границах, но это зависит от конкретной ситуации и требований к программе.

Если диапазон имеет большую длину и проверка в середине требует значительных ресурсов, то может быть целесообразно проводить проверки только на границах. Однако, в этом случае необходимо убедиться, что все возможные входные значения попадают в один из классов эквивалентности или на границы, иначе программа может давать некорректный результат.

Если же диапазон небольшой или проверка в середине не требует значительных ресурсов, то рекомендуется проводить проверку для всех значений в диапазоне, чтобы убедиться в правильности работы программы в любых условиях.

Таким образом, выбор метода проверки зависит от конкретных условий и требований к программе.

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?

1. Открыть приложение Яндекс Аренда на мобильном устройстве.
2. Найти раздел «Показать на карте» и нажать на него.
3. Дождаться появления сообщения «Непредвиденная ошибка» и закрытия приложения.
4. Подключить мобильное устройство к компьютеру.
5. Открыть программу для разработки приложений, например, Android Studio.
6. В настройках мобильного устройства включить режим разработчика и включить отладку по USB.
7. В Android Studio выбрать пункт «Run» -> «Debug app» и выбрать приложение Яндекс Аренда.
8. Дождаться загрузки приложения и повторить действия, которые приводят к появлению ошибки.
9. В Android Studio выбрать пункт «Logcat» и найти сообщение об ошибке.
10. Скопировать сообщение об ошибке и отправить его разработчику вместе с описанием действий, которые приводят к ошибке.