# Performance Review. Сентябрь 2020

## Список вопросов JS:

|  |  |  |  |
| --- | --- | --- | --- |
| Номер | Компетенция | Навык | Готово |
|  | ООП (знание парадигмы программирования) | Что такое Объект, класс, абстрактный класс и интерфейс? | ✓ |
|  | ООП (знание парадигмы программирования) | Что такое реализация и наследование | ✓ |
|  | ООП (знание парадигмы программирования) | Трейты - что такое? | ✓ |
|  | ООП (знание парадигмы программирования) | Абстракция | ✓ |
|  | ООП (знание парадигмы программирования) | Инкапсуляция | ✓ |
|  | ООП (знание парадигмы программирования) | Полиморфизм | ✓ |
|  | ООП (знание парадигмы программирования) | Где сильнее связанность при наследовании или инкапсуляции? | ✓ |
|  | ООП (знание парадигмы программирования) | Что такое прототип? | ✓ |
|  | ООП (знание парадигмы программирования) | function getSum(){return this.sum}. От куда возьмётся this? Как его можно поменять? В каких случаях this не определён? | ✓ |
|  | Объектно-ориентированный дизайн (SOLID) | Показать как минимум 2 примера использования принципа единой ответсвенности в своём коде, который был написан в период с 01.07.2020 по настоящее время | ✓ |
|  | Объектно-ориентированный дизайн (SOLID) | Показать как минимум 2 примера использования принципа открытости/закрытости в своём коде, который был написан в период с 01.07.2020 по настоящее время, доказать, что принцип соблюдается т.е. он пригодился хотя бы 1 раз | ✓ |
|  | Объектно-ориентированный дизайн (SOLID) | Показать как минимум 2 примера использования принципа подстановки Барбары Лисков, который был написан в период с 01.07.2020 по настоящее время, обсоновать, что он уже сейчас работает. | ✓ |
|  | Объектно-ориентированный дизайн (SOLID) | Показать как минимум 2 примера использования принципа разделения интефейсов, которые были написаны в период с 01.07.2020 по настоящее время. | ✓ |
|  | Объектно-ориентированный дизайн (SOLID) | Показать как минимум 2 примера использования принципа инверсии зависимостей, которые были написаны в период с 01.07.2020 по настоящее время. Кто является заказчиком данных частей, почему мы считаем это принципом единой отвесвенности. | ✓ |
|  | Объектно-ориентированный дизайн (SOLID) | Когда необходимо применять принцип обратной инверсии зависмости? | ✓ |
|  | Знание HTML | Необходимо сверстать страницу для не видящих людей. Как это сделать? | ✓ |
|  | Знание HTML | Необходимо сверстать страницу как для видящих, так для не видящих. Как это сделать? | ✓ |
|  | Знание HTML | Сверстать флаг российской федерации во всё окно браузера с равными пропорциями. | ✓ |
|  | Знание CSS | Сверстать флаг российской федерации во всё окно браузера с равными пропорциями. | ✓ |
|  | Знание CSS | Есть ли возможность изменить цвет svg картинки? | ✓ |
|  | Знание CSS | Как в css можно анимировать переход от одного состояния в другое? | ✓ |
|  | Знание CSS | Как распределить элементы равной ширины, чтобы между ними было равное растояния, зависящее от ширины родительского блока? | ✓ |
|  | Знание JS | Сделайте POST запрос к серверу из браузера не используя библиотеки? | ✓ |
|  | Знание JS | Какие проблемы могут возникнуть при запросе другой домен? | ✓ |
|  | Знание JS | Только неоптимальные запросы могут быть проблемой производительности БД? | ✓ |
|  | Знание JS | Видны ли переменные var в замыкании? | ✓ |
|  | Знание алгоритмов и структур данных | Есть 2-х мерный массив 3 на 3 каждый элемент - пустая строка, символ '0' или символ 'X', определить является ли данная позиция выйгрышем в крестики нолики? Кто выйграл, если позиция игра выйграна? Ошибка, если выйгрыш 2 раза и более. | ✓ |
|  | Декомпозиция задач | Как можно разбить большую задачу, чтобы заказчик мог проверять степень её выполнения? | ✓ |
|  | Декомпозиция задач | Нужен визуальный редактор диаграммы ганта. Какие вопросы надо задать и какие подзадачи можно выделить? | ✓ |
|  | Умеет самостоятельно проводить код-review | Тестовое задание | ✓ |
|  | Знание Git | Какими коммандами отправить изменения, которые вы только что внесли, в репозиторий? | ✓ |
|  | Знание Git | Какими командами можно получить изменения в текущей ветки из репозитория | ✓ |
|  | Знание Git | Как переключиться на ветку 'test', которая есть в репозитории origin, но нет локально? | ✓ |
|  | Знание Git | Вы сделали коммит не в ту ветку, в которую хотели сделать коммит. Как переместить коммит в другую ветку? | ✓ |
|  | Знание Git | Как отменить коммит? | ✓ |
|  | Знание unit - тестов | Примеры написанных тестов. | ✓ |
|  | Знание unit - тестов | Примеры тестов с подменой зависимостей. |  |
|  | Знание технического английского | [https://github.com/strongloop/strong-soap Как можно получить soap-заголовки?](https://github.com/timiki/rpc-server-bundle) | ✓ |
|  | Знание React | Написать компонент для галочки, галочка должна выглядеть одинаково в разных местах. Компонент должен метод, который будте вызван при изменении состояния. | ✓ |
|  | Декомпозиция больших задач, фич, их оценка и распределение по разработчикам, разработка архитектуры | Компания по производству мороженного в Екатеринбурге решила открыть свой интернет магазин для розницы. Какие вопросы необходимо решить для начала разработки? | ✓ |
|  | Знание сторонних библиотек | Какие библиотеки/фреймворки знаете? | ✓ |
|  | Анализ сбоев | Какие случаи были за последнии полгода. Приложите задачу. |  |
|  | Улучшение действующей архитектуры кода | Примеры. Приложите ссылку на коммит. | ✓ |
|  | Изучение новых библиотек | Какую бибилотеку вы ппробовали изучить за последнии 3 месяца? | ✓ |
|  | Модули | Что такое принцип эквивалентности повторгного использования и выпусков? | ✓ |
|  | Модули | Что такое принцип согласованного изменения? | ✓ |
|  | Модули | Что такое принцип совместного повторного использования? | ✓ |
|  | Модули | Противоречать ли принциы "эквивалентности повторгного использования и выпусков", "принцип согласованного изменения" и "совместного повторного использования" друг другу? | ✓ |
|  | Качество кода | и 5 коммитов которыми гордитесь, если нет таких, то за которые не стыдно. | ✓ |

## Что такое Объект, класс, абстрактный класс и интерфейс?

Объект – это тип данных, который используется для хранения коллекций различных значений и сложных сущностей.

Класс – это сущность, описывающая какой-либо абстрактный тип данных.

В JavaScript класс – это разновидность функции.

Абстрактный класс – класс, определяющий абстрактные методы, от которого будут наследоваться классы, а не создаваться экземпляры.

Интерфейс – контракт класса реализовывать определенные методы.

Интерфейс – внутренний и внешний, методы и свойства класса.

## Что такое реализация и наследование.

Реализация интерфейса – полностью определенный класс.

Наследование – концепция при которой абстрактный класс может наследовать методы и свойства другого класса.

## Трейты - что такое?

Трейт – механизм позволяющий группировать методы и наследовать их.

## Абстракция.

Абстракция — это объявление только тех методов и свойств которые необходимы в исследуемой области.

## Инкапсуляция.

Инкапсуляция — это размещение в одном месте свойств и методов управления ими, а также защита их от использования извне.

## Полиморфизм.

Полиморфизм – способность метода работать с разными типами данных

## Где сильнее связанность при наследовании или инкапсуляции?

Связанность сильнее при наследовании исходя из определения инкапсуляции. При наследовании компоненты (классы) связаны и зависят друг от друга в этом случае сильнее.

## Что такое прототип?

Экземпляр объекта, который скопировали, для наследования методов и свойств.

## function getSum(){return this.sum}. От куда возьмётся this? Как его можно поменять? В каких случаях this не определён?

this это ссылка на объект который вызывает getSum().

Если функция была вызвана без объекта, то this== undefined

## Показать, как минимум 2 примера использования принципа единой ответственности в своём коде, который был написан в период с 01.07.2020 по настоящее время.

Каждый программный объект имеет одно и только одно назначение

<https://github.com/ubrr/catalog-manager/blob/master/src/Entity/PassportRecord.php>

https://github.com/ubrr/catalog-manager/blob/master/assets/js/hooks/http.hook.js

## Показать, как минимум 2 примера использования принципа открытости/закрытости в своём коде, который был написан в период с 01.07.2020 по настоящее время, доказать, что принцип соблюдается т.е. он пригодился хотя бы 1 раз.

Класс открыт для расширения, но закрыт для модификации

<https://github.com/ubrr/catalog-manager/blob/master/src/Core/Condition.php>

https://github.com/ubrr/catalog-manager/blob/master/src/Core/Reference.php

## Показать, как минимум 2 примера использования принципа подстановки Барбары Лисков, который был написан в период с 01.07.2020 по настоящее время, обосновать, что он уже сейчас работает.

Если у нас есть класс A (не виртуальный, а вполне реально используемый в коде) и отнаследованный от него класс B, то если мы заменим все использования класса A на B, ничего не должно измениться в работе программы. Ведь класс B всего лишь расширяет функционал класса A.

## Показать, как минимум 2 примера использования принципа разделения интерфейсов, которые были написаны в период с 01.07.2020 по настоящее время.

Клиенты не должны реализовывать методы, которые им не нужно использовать. Кроме того, этот принцип указывает на то, что интерфейс должен решать лишь какую-то одну задачу (в этом он похож на принцип единственной ответственности), поэтому всё, что выходит за рамки этой задачи, должно быть вынесено в другой интерфейс или интерфейсы.

## Показать, как минимум 2 примера использования принципа инверсии зависимостей, которые были написаны в период с 01.07.2020 по настоящее время. Кто является заказчиком данных частей, почему мы считаем это принципом единой ответственности.

Модуль это элемент декомпозиции системы.

Модули верхних уровней не должны завесить от модулей нижних.

Инверсия зависимостей применяется когда много модулей и некоторые модули зависят от модулей которые часто меняются.

Могут возникать ошибки при использовании модулей которые поменялись.

Для решения этой проблемы используются интерфейсы адаптеры которые инкапсулирут используемые методы

Принцип единой ответсвенности потому что модули которые ты меняешь иоогут быть использованы другими модулями

https://github.com/ubrr/catalog-manager/blob/master/src/Controller/

## Когда необходимо применять принцип обратной инверсии зависимости?

Может быть когда нужна более сильная связь между модулями и модули при этом не будут часто меняться

## Необходимо сверстать страницу для не видящих людей. Как это сделать?

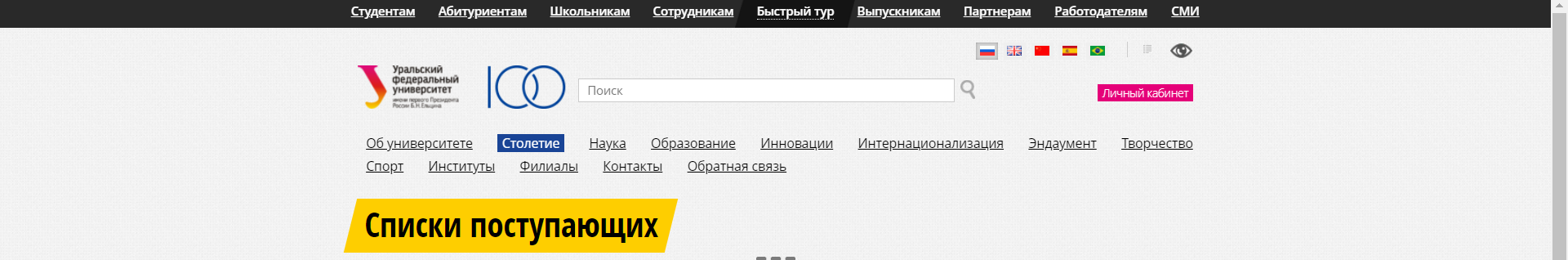
Для этого необходимо учитывать принципы ux ui при построении интерфейса:

* Читабельность текста: размер, сочетание цветов, шрифт.
* Контрастные блоки с ключевой информацией
* Четкое логически выверенное построение информации на странице

Так же необходимо добавить возможность использования клавиатуры для управления, и голосовое сопровождения для воспроизведения активного блока

## Необходимо сверстать страницу как для видящих, так для не видящих. Как это сделать?

Для того чтобы страница была адаптирована под любую категорию можно сделать переключатель для слабовидящих людей который будет настраивать страницу под определенные пользовательские проблемы



## Сверстать флаг российской федерации во всё окно браузера с равными пропорциями.

Папка flag

## Сверстать флаг российской федерации во всё окно браузера с равными пропорциями.

Папка flag

## Есть ли возможность изменить цвет svg картинки?

Папка svgcolor

Можно менять цвет при некоторых вариантах вставки svg

https://github.com/andrysazonov/smap

## Как в css можно анимировать переход от одного состояния в другое?

С помощью свойств:

* transition-property
* transition-duration
* transition-timing-function
* transition-delay

Указываем свойство анимацию и как будет меняться  
При изменении этого свойства происходит анимация

## Как распределить элементы равной ширины, чтобы между ними было равное растояния, зависящее от ширины родительского блока?

У родительского блока задать свойство

Displyay: flex;

Justify-content:space-between

## Сделайте POST запрос к серверу из браузера не используя библиотеки?

В js сделать http запросы можно двумя способами.

1 способ

XMLHttpRequest – это встроенный в браузер объект, который даёт возможность делать HTTP-запросы к серверу без перезагрузки страницы.

let xhr = new XMLHttpRequest();

xhr.open(method, URL, [async, user, password])

xhr.send([body])

2 способ

C Помощью Функции Fetch

let promise = fetch(url, [options])

let user = {

name: 'John',

surname: 'Smith'

};

let response = await fetch('/article/fetch/post/user', {

method: 'POST',

headers: {

'Content-Type': 'application/json;charset=utf-8'

},

body: JSON.stringify(user)

});

let result = await response.json();

alert(result.message);

## Какие проблемы могут возникнуть при запросе другой домен?

При запросе на другой домен может сработать политика CORS при которой запрос не будет обработан

Cross-Origin Resource Sharing (CORS) — механизм, использующий дополнительные HTTP-заголовки, чтобы дать возможность агенту пользователя получать разрешения на доступ к выбранным ресурсам с сервера на источнике (домене), отличном от того, что сайт использует в данный момент.

Если в ресурсе на который посылается cross-origin HTTP request, политикой безопасности определено что не все могут посылать кросс доменные запросы, то ответ будет отправлен обратно.

Для этого в заголовке ответа устанавливаются определенные парамтеры.

## Только неоптимальные запросы могут быть проблемой производительности БД?

Нет, так же БД должна удовлетворять 3 нормам построения.

Атомарность, отсутствие зависимости не ключевых полей от части составного ключа. отсутствие зависимостии не ключевых полей от других не ключевых полей

## Видны ли переменные var в замыкании?

Видны.

У каждого блока кода есть скрытый объект Лексическое окружение, которое состоит из двух свойств

1. Environment Record – объект, в котором как свойства хранятся все локальные переменные (а также некоторая другая информация, такая как значение this).
2. Ссылка на внешнее лексическое окружение – то есть то, которое соответствует коду снаружи (снаружи от текущих фигурных скобок).

Когда функция хочет получить доступ к переменной, она берёт её текущее значение из своего или внешнего лексического окружения.

## Есть 2-х мерный массив 3 на 3 каждый элемент - пустая строка, символ '0' или символ 'X', определить является ли данная позиция выйгрышем в крестики нолики? Кто выйграл, если позиция игра выйграна? Ошибка, если выйгрыш 2 раза и более.

Папка tic tac toe

## Как можно разбить большую задачу, чтобы заказчик мог проверять степень её выполнения?

Если разработка продукта ведется по методологии Agile и Фреймворку Scrum то подразумевается, что в конце каждого спринта получается инкремент продукта, следовательно заказчик может оценивать продукт в актуальном виде.

Разбивание задачи при этом происходит в зависимости от нововведений в продукт

Так же большую задачу разбить на подзадачи, чтобы заказчик мог проверять степень выполнения, можно по функционалу, то есть каждая подзадача это маленький функционал, работу которого можно оценить.

## Нужен визуальный редактор диаграммы ганта. Какие вопросы надо задать и какие подзадачи можно выделить?

Распределение ролей в использовании редактора

Используемые цвета, дизайн

Логику механику взаимодействия с задачами

Возможность масштабируемости диаграммы

Максимальное количество выводимых задач

Максимальный срок по которым выводятся задачи

Сколько человек будет использовать диаграмму

Количество видов задач

Детализация задачи (описание, авторы …)

Возможность связывать задачи

Откуда будут браться данные для диаграммы

Возможно ли использование существующих аналогов или придется создать свою

Подзадачи при разработке:

Разработка дизайна

Разработка обмена данными с хранилищем

Разработка механики взаимодействия

итд

## Тестовое задание

Умеет ли сам проводить code review. Умею

## Какими коммандами отправить изменения, которые вы только что внесли, в репозиторий?

Git add .

Git commit –m”сообщение коммита”

Git push «удаленный репозиторий» «ветка»

## Какими командами можно получить изменения в текущей ветки из репозитория

Git pull

Git fetch +git merge

## Как переключиться на ветку 'test', которая есть в репозитории origin, но нет локально?

Git pull

Git checkout test

## Вы сделали коммит не в ту ветку, в которую хотели сделать коммит. Как переместить коммит в другую ветку?

Git checkout «желаемая ветка»

Git cherry-pick « ветка куда влили»

## Как отменить коммит?

1 способ

Изменить файлы и сделать коммит, который будет отменять предыдущие изменения

Git revert head

2 способ

git reset --hard v1

## Примеры написанных тестов.

https://github.com/ubrr/catalog-manager/tree/master/src/test/PassportReference

## Примеры тестов с подменой зависимостей.

## https://github.com/strongloop/strong-soap Как можно получить soap-заголовки?

Received SOAP headers

A service method can look at the SOAP headers by providing a third arguments.

{

HeadersAwareFunction: function(args, cb, headers) {

return {

name: headers.Token

};

}

}

It is also possible to subscribe to the 'headers' event. The event is triggered before the service method is called, and only when the SOAP Headers are not empty.

server = soap.listen(...)

server.on('headers', function(headers, methodName) {

// It is possible to change the value of the headers

// before they are handed to the service method.

// It is also possible to throw a SOAP Fault

});

First parameter is the Headers object; second parameter is the name of the SOAP method that will called (in case you need to handle the headers differently based on the method).

## Написать компонент для галочки, галочка должна выглядеть одинаково в разных местах. Компонент должен метод, который будте вызван при изменении состояния.

Папка checkbox

## Компания по производству мороженного в Екатеринбурге решила открыть свой интернет магазин для розницы. Какие вопросы необходимо решить для начала разработки?

Вопросы можно разделить на организационные и технические

Технические:

* Разработка шаблонного интернет магазина или разработка дизайна
* Стек технологий
* Какая товароучетка будет подключена(интеграция с CRM)
* Будет ли оплата(эквайринг)
* Какие будут дополнительные фишки на сайте (бот помощник, реклама, видео, аналитика)
* CEO продвижение
* Панель администратора
* Архитектуру веб приложения
* Архитектуру сайта (категории, страницы …)
* Возможность авторизации клиентов
* Информационная безопасность

Возможности для поддержки

Организационные:

* Продумать логистику
* Продумать кто будет заниматься поддержкой и администрированием сайта
* Кто целевая аудитория сайта и нужен ли он
* Возможные точки роста при создании сайта
* Продумать над тем, где взять инвестиции для сайта
* Бонусная программа для клиентов

итд

## Какие библиотеки/фреймворки знаете?

* Material ui
* Materialize
* Bootstrap
* bootstrap react
* react
* jquery
* styled-components
* webpack
* momento
* rebass

## Какие случаи были за последнии полгода. Приложите задачу.

## Примеры. Приложите ссылку на коммит.

Задачи где использовал webpack для сборки компонента,

Build form as component, calculator …

## Какую бибилотеку вы пробовали изучить за последнии 3 месяца?

* Momento
* webpack

## Что такое принцип эквивалентности повторного использования и выпусков?

Компонент должен быть целостным и решать какую то цель

## Что такое принцип согласованного изменения?

Это как принип единой ответственности только для компонентов

Значит что классы и объекты, свойства которые меняются в один момент должны помещяться в один компонент

К примеру не надо включать в один компонент, обработчик другого

## Что такое принцип совместного повторного использования?

Это как принцип разделения интерфейсов, только для компонентов

Классы без связи не включаются в один компонент

Как компонент в реакте

Компонент Slider должн быть только двигаться и не отвечать за какой нибудь попап

## Противоречать ли принциы "эквивалентности повторгного использования и выпусков", "принцип согласованного изменения" и "совместного повторного использования" друг другу?

Противоречат друг другу

принцип согласованного изменения и принцип эквивалентности повторного использования и выпусков описывают подход когда компонент это законченный объект с функцией и обработкой и содержит логику его касающуюся

А принцип совместного повторного использования наоборот говорит о том, что компонент должен включат только конкретно его функционал и ничего больше

## и 5 коммитов которыми гордитесь, если нет таких, то за которые не стыдно.

Catalog manager

Calculator-form

Calculator -cc -120-240

Build form as component

## Сделайте POST запрос к серверу из браузера не используя библиотеки?

<https://learn.javascript.ru/xmlhttprequest>

<https://learn.javascript.ru/fetch>

## Видны ли переменные var в замыкании?

<https://learn.javascript.ru/closures>

<https://developer.mozilla.org/ru/docs/Web/JavaScript/Closures>