FUNBOX

Квалификационные задания для разработчиков JavaScript

О заданиях

Данные задания были разработаны для того, чтобы мы могли оценить уровень ваших знаний. Они составлены таким образом, что вы можете выполнить их в любое удобное для вас время в обстановке, которую считаете оптимальной.

Задания разделены на 2 категории:

Level I содержит вопросы, которые помогут вам продемонстрировать свой опыт и знание теоретических основ JavaScript.

Level IIсодержит задание, которое поможет вам продемонстрировать свои навыки на практике.

Результат присылайте нам на почтовый ящик wanted@fun-box.ru с темой

«Вакансия разработчика JavaScript [Senior/Middle/Junior]». После проверки заданий мы обязательно сообщим вам о нашем решении.

# Level I

Здесь несколько теоретических вопросов, c помощью которых вы сможете в свободной форме рассказать нам о своем личном опыте в том или ином аспекте, а также проиллюстрировать рассказ примерами своих работ.

## Q1

Расскажите, чем, на ваш взгляд, отличается хорошее клиентское приложение от плохого с точки зрения

* пользователя;
  + Интуитивно понятно, как управлять: «что нажимать, куда двигать …»
  + Работает быстро, не виснет, «летает»
  + Работает на любых устройствах
* менеджера проекта;
  + Модульность: легко расширять, модифицировать, находить и исправлять ошибки.
  + Масштабируемость по «объему»: числу пользователей, трафику,
  + Число пользователей растет экспоненциально.
  + От клиентов идут только восторженные отклики и предложения по развитию.
  + Код документирован и в минимальной степени («вообще!») не зависит от конкретных программистов.
  + Архитектура и технические решения, выбранные для приложения, не потребуют перехода на принципиально другой продукт в течение продолжительного времени.
* дизайнера;
  + Идеальное сочетание цветов, стилей,
  + Реалистичность картинки
  + Сногсшибательные визуальные эффекты (но к месту, без перегибов)
* верстальщика;
  + Минимальное число страниц
  + Все модификации минимально затрагивают верстку, добавления легко встраиваемы в наработанные шаблоны
* серверного программиста.
  + Все задачи от Менеджера проекта могут быть выполнены настройками в имеющемся коде.
  + В решении используются самые передовые инструменты и разработки.

## Q2

Опишите основные особенности разработки крупных многостраничных сайтов, функциональность которых может меняться в процессе реализации и поддержки.

Расскажите о своем опыте работы над подобными сайтами: какие подходы, инструменты и технологии вы применяли на практике, с какими проблемами сталкивались и как их решали.

Личного опыта в разработке многостраничных сайтов ку меня нет.

Думаю, что идеальным было бы разнесение работы над сайтом по идеологии, например, MVC. Это должно позволить сепарировать и локализовать модификации предметной области (модели), пользовательского интерфейса и визуализации сайта. В свою очередь, это позвлит разделит ьработу между спецаилистами с разными компетенциями.

## Q3

При разработке интерфейсов с использованием компонентной архитектуры часто используются термины **Presentational Сomponents** и **Сontainer Сomponents**. Что означают данные термины? Зачем нужно такое разделение, какие у него есть плюсы и минусы?

#### *Насколько я понял, эти термины пришли из React. До Вашего вопроса я не разбирался с этим. Хотя, когда стал решать практическое задание, то увидел, что этот подход используют разработчики, например, в react-yandex-maps или react-beautiful-dnd. Но пока я его до конца ещё не освоил.*

Пока же моё представление такое:

Container —react- компонент, в котором принимают и работают с данными. Т.е. это ближе к C в MVC/

Presentational — react-компонент, в котором отображают данные. А это V в МVС.

Преимущества указывают такие:

* Разделение сущностей: что есть интерфейс, а что «внутренние дела» приложения.
* Повторное использование. Презентационные компоненты можно стыковать с разными управляющим компонентами («состояниями»), если требуется такой же функционал
* Работу с презентационным компонентом можно передать другим специалистам: дизайнерам, тестировщикам скриншотов, ... При этом не будет затронута логика внутренней обработки.

Про минусы и сложности могу только предположить, что надо делить «на два куска» то, что привычно писалось в одном. В каких-то простых случаях это, наверное, может привести к увеличению числа модулей (файлов) и величины кода.

## Q4

Как устроено наследование в JS? Расскажите о своем опыте реализации JSнаследования без использования фреймворков.

Наследование устроено через специальную ссылку \_\_proto\_\_. В ней записан класс родителя. Через неё у потомка есть доступ к методам родителя. Потомок модет переопределить класс родителя.

Я использовал наследование в проекте Шахарды. О нем есть информация вот здесь: <https://hvl-vc.000webhostapp.com/today/shahards/>. А исходные файлы – вот здесь: <https://github.com/VladimirLHk/shahards/tree/master/interface/src>.

Главным родительским классом был GameTreasurer. В нем хранились методы, реализующие правила игры.

От него наследовали три класса:

* «Администратор» (GameMaster)— ведет игру: организует правильную последовательность действий в игре,
* «Секретарь» (GameSecretary)— ведет протокол, фиксирует всю значимую информацию о ходе игры,
* «Игрок-робот» (PlayerBot) — делает шаги.

## Q5

Какие библиотеки можно использовать для написания тестов end-to-end во фронтенде? Расскажите о своем опыте тестирования веб-приложений.

С тестированием я только начинаю знакомиться.

Из того, что быстро удалось увидеть в интернете:

* Casper.js,
* CodeceptJS
* NightmareJS,
* phantom.js,
* Protractor,
* Selenium
* watir,
* webdriverio,

## Q6

Вам нужно реализовать форму для отправки данных на сервер, состоящую из нескольких шагов. В вашем распоряжении дизайн формы и статичная верстка, в которой не показано, как форма должна работать в динамике. Подробного описания, как должны вести себя различные поля в зависимости от действий пользователя, в требованиях к проекту нет. Ваши действия?

Если доработать требования к проекту – это не вариант, то надо залазить в «предметную область»: что за данные передаются на сервер, какие между ними могут быть зависимости-ограничения.

## Q7

Расскажите, какие инструменты помогают вам экономить время в процессе написания, проверки и отладки кода.

Мой самый главный инструмент - это PHPStorm: подсказывает, проверяет, запускает localhost, …

Часто прибегаю к DevTools Chrom: посмотреть оформление элементов верстки, пошагово пройти по коду, …

## Q8

Какие ресурсы вы используете для развития в профессиональной сфере? Приведите несколько конкретных примеров (сайты, блоги и так далее).

По JS я учился и теперь использую как справочник-напоминалку сайт Ильи Кантора: <https://learn.javascript.ru/>.

По HTML и CSS часто обращаюсь к: <https://www.w3.org/> <https://developer.mozilla.org/> <http://htmlbook.ru/>

По PHP я учусь по книге PHP 7 И.Симдянова и Д. Котерова, смотрю <http://php.net/>

Очень интеерсный «сайт-задчник» белорусского разработчика Дмитрия Трепачева: <http://code.mu>

Под конкретные вопросы и темы использую: <https://habr.com/r>, <https://tproger.ru>, <https://medium.com/>, [https://ru.stackoverflow.com/](https://ru.stackoverflow.com/б), [www.youtube.com](http://www.youtube.com)

Какие ещё области знаний, кроме тех, что непосредственно относятся к работе, вам интересны?

Генеалогия, музыка и иностранные языки. Более подробно про это:

<https://hvl-vc.000webhostapp.com/future/not-job/>

<https://hvl-vc.000webhostapp.com/future/50-50/>

## Q9

Расскажите нам немного о себе и предоставьте несколько ссылок на последние работы, выполненные вами.

Последние 18 лет я занимался управленским и владельческим консалтингом. Приход в эту облазь был стечением обстоятельств. Нельзя сказать, что я бредил этой профессией. Все-таки мне ближе более «точные науки». Со временим у меня стала копиться неудовлетворенность тем, чем я занимаюсь. В итоге я решил заняться тем, к чему более приспособлен мой мозг: я решил вернуться в программирование.

Обо всем этом более подробно я написал здесь: <https://hvl-vc.000webhostapp.com/hard-questions/>

Про мои последние работы в программировании можно посмотреть вот здесь: <https://hvl-vc.000webhostapp.com/today/>

## Level II

На этом уровне находится практическое задание.

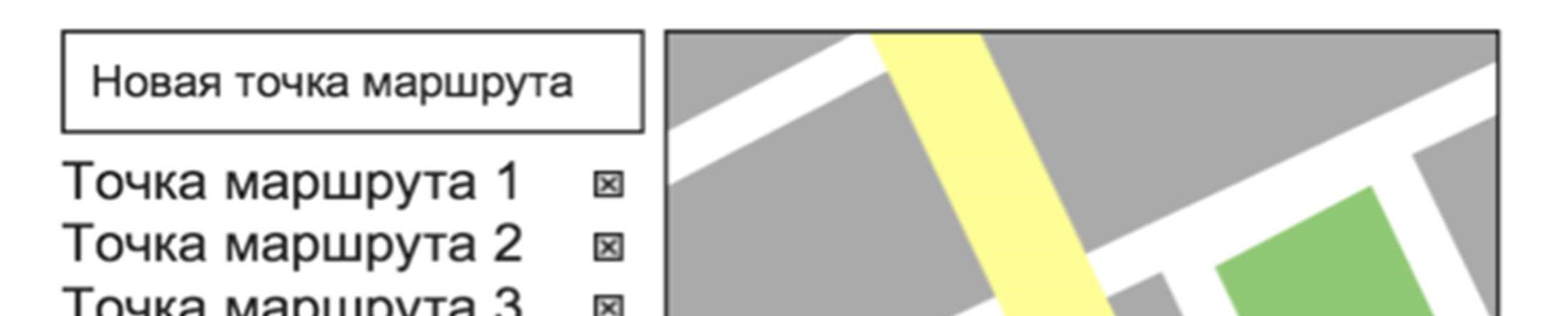
### Task Description

Необходимо реализовать редактор маршрутов — одностраничное приложение, в котором пользователь в интерактивном режиме может создавать на карте маршрут, указывая начальную, конечную и промежуточные точки движения. Для каждой точки маршрута можно посмотреть ее адрес.

Приложение визуально состоит из:

 текстового поля ввода для новых точек маршрута;  списка уже введенных точек маршрута;  интерактивной карты.

На картинке ниже показан схематичный пример, итоговый внешний вид может быть любым, на ваше усмотрение. Приложение должно адекватно вести себя при изменении размера окна браузера, взаимное расположение элементов не должно изменяться.



### Task Functional Requirements

Новая точка маршрута добавляется с помощью ввода ее названия в текстовом поле и нажатия Enter. После этого:

* введенная точка маршрута отображается в конце списка уже добавленных точек;
* в текущем центре карты появляется маркер, обозначающий новую точку маршрута.

Напротив каждой точки маршрута в списке находится кнопка удаления, при ее нажатии точка маршрута пропадает из списка, а с карты пропадает ее маркер.

Порядок точек маршрута в списке можно изменять перетаскиванием.

Маркеры, соответствующие точкам маршрута, можно перемещать по карте перетаскиванием.

Маркеры на карте соединены прямыми линиями в том порядке, в котором они находятся в списке. Полученная таким образом ломаная изображает маршрут, первая точка в списке — начало маршрута, последняя — конец маршрута.

При изменении порядка точек в списке или их удалении, а также при перемещении маркеров маршрут на карте автоматически перерисовывается.

При клике на маркер появляется балун, в балуне отображается название соответствующей ему точки.

### Task Technical Requirements

* Яндекс.Карты или Google Maps.
* React.js.
* Покрытие полученного кода тестами. Этого пока нет
* Ссылка на страницу с результатом <https://hvl-items.000webhostapp.com/route-editor/>   
   и git-репозиторий с исходниками <https://github.com/VladimirLHk/Route-Editor> .
* Инструкция для запуска проекта.

Не совсем понял, что понимается под запуском. Если демонстрация, то она сама откроется по ссылке. Если сборка из исходников, то нужны модули react, eact-dom, react-beautiful-dnd, react-yandex-maps

Благодарим вас за время, потраченное на выполнение заданий.