

Тестовые задания ШРИ 2018-II:

Задание №1 «Комплексное» («Найди ошибки»)

Представьте, что вы инженер в большом интернет-магазине и отвечаете за доставку заказов дронами.

В каждый момент времени в воздухе находится несколько сотен дронов. Для управления ими развёрнута сеть базовых станций. В ваших обязанностях — контроль работоспособности станций и устранение неисправностей. Для этого есть специальное приложение, в котором на карте города показаны места размещения базовых станций и информация о них.

Репозиторий на GitHub: <https://github.com/yandex-shri-2018/entrance-task-1-2>

К сожалению, после очередного релиза вашего приложения данные на карте перестали отображаться. Кажется, программисты опять понаделали ошибок в коде. Нужно скорее найти ошибки и починить их.

Описание:

- Вся область экрана занимает интерактивная карта Москвы.
- На карте отображаются места размещения базовых станций.
- Если на небольшом пространстве много объектов, они объединяются в кластер.
- При клике на кластер карта масштабируется для просмотра объектов, входящих в него.
- Неисправные станции обозначаются на карте красным цветом, исправные — синим.
- Используя фильтр, можно отобразить на карте объекты с нужным состоянием — например, отобразить только неисправные.
- Если неисправный объект входит в кластер, то иконка кластера должна показывать, что в нем есть неисправная станция.
- При клике на метку базовой станции появляется попап с информацией о ней: серийный номер, состояние, количество активных дронов, график нагрузки.

Техническое описание:

- Приложение работает в браузере и написано на JavaScript, модули собираются с помощью Webpack (<https://webpack.js.org>)
- Для отображения карты используется API Яндекс.Карт. (<https://tech.yandex.ru/maps/isapi/>)
- Для отображения графиков используется Chart.js (<https://www.chartjs.org>)

Что мы проверяем этим заданием:

В этом задании мы хотим проверить вашу способность разбираться в незнакомом коде и API, а также ваш навык отладки. Пожалуйста, опишите в коде или файле README ход ваших мыслей: как и какие ошибки вы нашли, почему они возникли, какие существуют способы их исправления. Мы не ограничиваем вас в использовании сторонних инструментов и библиотек, но будем ждать от вас комментария — что и зачем вы использовали.

Задание № 2 На верстку

Это задание проверяет, насколько хорошо вы умеете верстать и знаете особенности браузеров.

Вам нужно сверстать панель управления «умным домом». Она показывает информацию о текущем состоянии дома и позволяет управлять устройствами вручную и по сценариям. Дизайнер подготовил для вас отдельные макеты для большого экрана и мобильных телефонов.

Макет: <https://yandex-shri-2018.github.io/entrance-task-2-2>

Репозиторий на GitHub: <https://github.com/yandex-shri-2018/entrance-task-2-2>

Компоновка страницы

- В верхней части страницы есть шапка. В ней находится логотип и главное меню. Для адресов ссылок используйте заглушку #.
- В нижней части находится подвал. Он должен примыкать к нижней границе экрана, вне зависимости от количества контента на странице.
- Вертикальный скролл есть только в мобильной версии, при прокрутке шапка остаётся на месте. Главное меню прячется за иконку ☰.

Содержимое страницы

На странице отображаются три блока: «Главное», «Избранные сценарии», «Избранные устройства».

Главное

Блок «Главное» находится слева и занимает половину экрана. В нём в виде текста отображается состояние окон и дверей, температура в доме и на улице. Справа находится список ближайших запланированных сценариев, который можно листать по вертикали. На мобильных устройствах блок занимает всю ширину экрана, а список ближайших сценариев листается по горизонтали.

Избранные сценарии

Правую половину экрана занимает блок «Избранные сценарии». Сами сценарии отображаются в виде панелей фиксированного размера, которые расположены в три ряда. Цвет иконки показывает, активен ли сценарий сейчас.

Если все сценарии не помещаются в три ряда, то справа сверху появляются кнопки для постраничного листания.

Переход между страницами должен быть анимированным. Вид и параметры анимации выберите самостоятельно. На мобильных устройствах вместо постраничного листания используется горизонтальная прокрутка списка.

Избранные устройства

Устройства, как и сценарии, отображаются в виде панелей фиксированного размера. Они расположены в один ряд по всей ширине экрана. Если устройства не помещаются, справа над списком появляются кнопки для постраничного листания — как в списке сценариев. На мобильных устройствах вместо постраничного листания используется горизонтальная прокрутка списка.

Список устройств можно фильтровать по комнатам и типам устройств. В мобильной версии фильтр отображается в виде меню. Для меню нет макета — реализуйте его на свое усмотрение.

При клике на устройство должен открываться попап управления им. Открытие попапа должно сопровождаться анимацией. Пример анимации есть в репозитории с макетами.

Яркость ламп и температура теплого пола выбираются при помощи слайдера. Термостатом можно управлять при помощи крутилки.

Критерии

- В первую очередь мы будем проверять, сверстаны ли страницы в точном соответствии с макетами.
- Вёрстка должна корректно выглядеть:
 - На десктопе — в последних версиях Google Chrome, Яндекс.Браузера, Mozilla Firefox, Safari, Microsoft Edge.
 - На мобильных устройствах — в Safari (iOS) и Google Chrome (Android).
- В этом задании мы проверяем ваши навыки вёрстки. Вы можете использовать JavaScript, если посчитаете это нужным, но старайтесь, чтобы код был как можно проще. Пожалуйста, не используйте JavaScript-фреймворки.
- По возможности используйте приёмы безопасной деградации CSS.
- Уделите внимание организации и оформлению кода. Оптимизация производительности и автоматизация будут плюсом.

Задание №3 на JS

Цель задания — реализовать алгоритм работы «умного дома», который будет производить расчёт стоимости потребляемой электроэнергии в день и возвращать рекомендованное расписание использования электроприборов оптимизируя денежные затраты.

На вход подаются данные о тарифах, электроприборах и их максимальной потребляемой мощности.

Тарифы — это периоды в сутках, для которых задана отдельная стоимость киловатт-часа.

Приборы — это набор подключенных к «умному дому» электроприборов, для которых известна потребляемая мощность, длительность цикла работы, а также время дня, когда они используется. Каждый прибор должен отработать один цикл в сутки. Максимально потребляемая мощность указывается в ватт-часах.

На выходе должно получиться суточное расписание включения электроприборов. Каждый прибор за сутки должен отработать один цикл, а суммарная стоимость потраченной электроэнергии должна быть минимальной.

- При значении mode — day период с 07:00 до 21:00.
- При значении mode — night период с 21:00 до 07:00 следующего дня.
- При значении mode — undefined период отсутствует, прибор может работать в любой промежуток времени.

Примеры входных и выходных данных:

<https://github.com/yandex-shri-2018/entrance-task-3-2>

В качестве необязательного задания предлагаем продумать и спроектировать сценарии обработки некорректных входных данных и системных ошибок.

Вы можете использовать любые технологии, фреймворки и библиотеки. Для каждого выбранного инструмента напишите небольшое обоснование — зачем он нужен в вашем проекте и почему именно он.

Мы будем оценивать реализацию функциональности по следующим критериям:

- Правильность алгоритма на разнообразных входных данных.
- Оформление кода
- Производительность
- Наличие и качество тестов