/\*Создайте массив, состоящий из 8 сгенерированных JS объектов, которые

будут описывать похожие объявления неподалёку \*/

// объявляю массив

var similarHotels = [];

//как забить массив заданными числовыми значениями

var arrayPush = function (min, max) {

    var newArray = [];

  for (var i = min; i<= max; i++) {

    newArray.push(i);

  }

    return newArray;

}

var numbersArray = arrayPush(1, 8);

//функция возвращающая случайное число из заданного диапазона

var renderRandomNumber = function (min, max) {

    var randomNumber = Math.floor(Math.random() \* (max - min + 1) + min);

    return randomNumber;

    }

//функция выбора случайного элемента из массива

var renderRandomArrayItem = function(array) {

    var i = renderRandomNumber(0, array.length-1);

    var randomItem = array[i];

    return randomItem;

}

// функция случайного перемещивания элементов массива

function shuffle(array) {

    var currentIndex = array.length, temporaryValue, randomIndex;

    while (0 !== currentIndex) {

      randomIndex = Math.floor(Math.random() \* currentIndex);

      currentIndex -= 1;

      temporaryValue = array[currentIndex];

      array[currentIndex] = array[randomIndex];

      array[randomIndex] = temporaryValue;

    }

    return array;

  }

//функция выдающая несколько случайных элементов из массива // случайным должно быть число окончания цикла. и до этого перемеживать надо

var pushRandomArrayItems = function (array) {

    shuffle(array);// перемешаем массив

    var randomArrayItems = '';

    for (var i = 0; i<= renderRandomNumber(0, array.length); i++) {

    randomArrayItems += array[i] +' ';

    }

    return randomArrayItems;

}

// все данные, которые будем использовать.

var TITLE\_DATA = ["Большая уютная квартира", "Маленькая неуютная квартира", "Огромный прекрасный дворец", "Маленький ужасный дворец", "Красивый гостевой домик", "Некрасивый негостеприимный домик", "Уютное бунгало далеко от моря", "Неуютное бунгало по колено в воде"];

var TYPE\_DATA = ['palace', 'flat', 'house', 'bungalo'];

var CHECKINOUT\_DATA = ['12:00', '13:00', '14:00'];

var FEATURES\_DATA = ["wifi", "dishwasher", "parking", "washer", "elevator", "conditioner"]; // надо сделать массив строк случайной длины из ниже предложенных

var PHOTOS\_DATA = ["http://o0.github.io/assets/images/tokyo/hotel1.jpg", "http://o0.github.io/assets/images/tokyo/hotel2.jpg", "http://o0.github.io/assets/images/tokyo/hotel3.jpg"];

//создадим функцию создания объектов с такими данными.

var generateHotelsArray = function () {

    var similarObject = {};

for (var i = 0; i < 8; i++) {

    similarObject = {

    author: {

        avatar: 'img/avatars/user' + '0' + numbersArray[i] + '.png'

    },

    offer: {

        title: shuffle(TITLE\_DATA)[i],

        address: renderRandomNumber(300, 900) + ', ' + renderRandomNumber(100, 500),

        price: renderRandomNumber(1000, 1000000),

        type: renderRandomArrayItem(TYPE\_DATA),

        rooms: renderRandomNumber(1, 5),

        guests: renderRandomNumber(1, 10), //????????????????????????

        checkin: renderRandomArrayItem(CHECKINOUT\_DATA),

        checkout: renderRandomArrayItem(CHECKINOUT\_DATA),

        features: pushRandomArrayItems(FEATURES\_DATA),

        description: '',

        photos: shuffle(PHOTOS\_DATA),

        location: {

            x: renderRandomNumber(300, 900),

            y: renderRandomNumber(130, 630)

        }

    }

    }

similarHotels.push(similarObject);

}

    return similarHotels;

}

similarHotels = generateHotelsArray();

console.log(similarHotels);

// 2 ЗАДАНИЕ

var map = document.querySelector('.map');

/\* map.classList.remove('map--faded'); \*/

//map\_\_pins - куда вставлять все ПИНы

var hotelsContainer = document.querySelector('.map\_\_pins');

//На основе данных, созданных в первом пункте, создайте DOM-элементы, соответствующие меткам на карте, и заполните их данными из массива.

//Итоговую разметку метки .map\_\_pin можно взять из шаблона .map\_\_card.

// элемент, КУДА будем копировать разметку из шаблона

var hotelPin = document.querySelector('.map\_\_pin');

// ЭТО ПИН

var pinTemplate = document.querySelector('template')

    .content

    .querySelector('.map\_\_pin');

// добавляю все это в функцию

var generatePinFromTemplate = function (hotel) {

    var hotelPinElement = pinTemplate.cloneNode(true);

    hotelPinElement.style.top = renderRandomNumber(100, 500) + 'px'; // ТАК ЗАДАЮТСЯ КООРДИНАТЫ!!!!!!!!!! задавать надо ПИНУ

    hotelPinElement.style.left = renderRandomNumber(300, 900) + 'px';

    var pinAvatar = hotelPinElement.querySelector('img');

    pinAvatar.src = 'img/avatars/user' + '0' + numbersArray[i] + '.png';

    pinAvatar.alt = TITLE\_DATA[i];

    return hotelPinElement;

}

//Отрисуйте сгенерированные DOM-элементы в блок .map\_\_pins. Для вставки элементов используйте DocumentFragment.

// ФУНКЦИЯ ДОБАВЛЕНИЯ В ВЕРСТКУ ЧЕРЕЗ ДОКУМЕНТ ФРАГМЕНТ

var fragment = document.createDocumentFragment();

    for (var i = 0; i < similarHotels.length; i++) {

        var newPin = generatePinFromTemplate(similarHotels[i]);

        fragment.appendChild(newPin);

    }

    hotelsContainer.appendChild(fragment);

    /\* var insertMapPins = function () {

        var fragment = document.createDocumentFragment();

    for (var i = 0; i < similarHotels.length; i++) {

        var newPin = generatePinFromTemplate(similarHotels[i]);

        fragment.appendChild(newPin);

    }

    hotelsContainer.appendChild(fragment);

    };

    insertMapPins(); \*/

// ОБЪЯВЛЕНИЯ

/\*.На основе ПЕРВОГО по порядку элемента ИЗ СГЕНЕРИРОВАННОГО МАССИВА и шаблона .map\_\_card создайте DOM-элемент объявления, заполните его

данными из объекта и вставьте полученный DOM-элемент

в блок .map перед блоком.map\_\_filters-container\*/

//Можно задать переменную для .map\_\_filters-container перед которым надо вставить элементы. И затем вместо аппенд использовать бифор

// map - КУДА ВСТАВЛЯТЬ

var mapFilter= document.querySelector('.map\_\_filters-container');

//ШАБЛОН, КОТОРЫЙ БУДЕТ КОПИРОВАТЬ -- ЭТО ОБЪЯВЛЕНИЕ

 var hotelTemplate = document.querySelector('template')

    .content

    .querySelector('.map\_\_card');

// по порядку задаю значения, но не добавляю никуда

var hotelElement = hotelTemplate.cloneNode(true);

var title = hotelElement.querySelector('h3');

title.classList.add('popup\_\_title');

title.textContent = similarHotels[0].offer.title;

var address = hotelElement.querySelector('p').querySelector('small');

address.classList.add('popup\_\_text--address');

address.textContent = similarHotels[0].offer.address;

var price = hotelElement.querySelector('.popup\_\_price');

price.classList.add('popup\_\_text--price');

price.textContent = similarHotels[0].offer.price + ' ₽/ночь';

var type = hotelElement.querySelector('h4');

type.classList.add('popup\_\_type');

var typeName = similarHotels[0].offer.type;

if(typeName == 'flat'){

    type.textContent ='Квартира';

}

if(typeName == 'bungalo'){

    type.textContent ='Бунгало';

}

if(typeName == 'palace') {

    type.textContent ='Дворец';

}

;

var capacity = hotelElement.querySelector('p:nth-of-type(3)');

capacity.classList.add('popup\_\_text--capacity');

capacity.textContent = similarHotels[0].offer.rooms + ' комната(ы) для ' + similarHotels[0].offer.guests + ' гост(я)ей';

var time = hotelElement.querySelector('p:nth-of-type(4)');

time.classList.add('popup\_\_text--time');

time.textContent = 'Заезд после ' + similarHotels[0].offer.checkin + ', выезд до ' + similarHotels[0].offer.checkout ;

/\* в шаблоне уже есть список с классами. сделать проверку, если в массиве нет значения Н, то удалить класс Н

\*/

var popupFeatures = hotelElement.querySelector('.popup\_\_features'); // нашел УЛ

var featureItems = similarHotels[0].offer.features.split(' '); //выводится массив со случайными значениями

console.log(featureItems);

//сматчить каждый элемент из списка ул

var wifiFeature = hotelElement.querySelector('.popup\_\_features').querySelector('li:nth-child(1)');

var dishwasherFeature = hotelElement.querySelector('.popup\_\_features').querySelector('li:nth-child(2)');

var parkingFeature = hotelElement.querySelector('.popup\_\_features').querySelector('li:nth-child(3)');

var washerFeature = hotelElement.querySelector('.popup\_\_features').querySelector('li:nth-child(4)');

var elevatorFeature = hotelElement.querySelector('.popup\_\_features').querySelector('li:nth-child(5)');

var conditionerFeature = hotelElement.querySelector('.popup\_\_features').querySelector('li:nth-child(6)');

console.log(wifiFeature);

if (featureItems.indexOf('wifi') == -1) {

    popupFeatures.removeChild(wifiFeature);

}

if (featureItems.indexOf('dishwasher') == -1) {

    popupFeatures.removeChild(dishwasherFeature);

}

if (featureItems.indexOf('parking') == -1) {

    popupFeatures.removeChild(parkingFeature);

}

if (featureItems.indexOf('washer') == -1) {

    popupFeatures.removeChild(washerFeature);

}

if (featureItems.indexOf('elevator') == -1) {

    popupFeatures.removeChild(elevatorFeature);

}

if (featureItems.indexOf('conditioner') == -1) {

    popupFeatures.removeChild(conditionerFeature);

}

var popupDescription = hotelElement.querySelector('p:nth-of-type(5)');

popupDescription.classList.add('.popup\_\_description');

popupDescription.textContent = similarHotels[0].offer.description;

var popupPhotos = hotelElement.querySelector('.popup\_\_pictures');

popupPhotos.classList.add('.popup\_\_photos');

popupPhotos.querySelector('li:nth-child(1)').querySelector('img').src = similarHotels[0].offer.photos[0];

popupPhotos.querySelector('li:nth-child(1)').querySelector('img').classList.add('popupPhoto');

popupPhotos.querySelector('li:nth-child(2)').querySelector('img').src = similarHotels[0].offer.photos[1];

popupPhotos.querySelector('li:nth-child(2)').querySelector('img').classList.add('popupPhoto');

popupPhotos.querySelector('li:nth-child(3)').querySelector('img').src = similarHotels[0].offer.photos[2];

popupPhotos.querySelector('li:nth-child(3)').querySelector('img').classList.add('popupPhoto');

popupPhotos.querySelector('.popupPhoto').naturalWidth = 10;

var popupAvatar = hotelElement.querySelector('img');

popupAvatar.src = similarHotels[0].author.avatar;

//выводим окно когда кликнут по МЕТКЕ СКОПИРУЮ ЭТО В СОБЫТИЕ КЛИКА

/\* map.insertBefore(hotelElement, mapFilter); \*/

//ЗАДАНИЕ 4

//АКТИВАЦИЯ страницы. по клику должна активироваться карта, появится метки, поле формы стать активным, адрес должен заполнится. и надо потом добавить проверку формы.

/\* Первое действие, которое нужно выполнить, перед тем, как приступить к этому

заданию, вернуть страницу в исходное состояние \*/

//добавить класс .map--faded. я просто в комментарии убрал в начале кода

console.log(map);

/\* поля формы должны быть неактивны в исходном состоянии. \*/

//добавил в верстке. удалять все равно потом через ДжиЭс

/\* 1. Активация страницы

Для этого нужно добавить обработчик

события mouseup на элемент .map\_\_pin--main.

Обработчик события mouseup должен вызывать функцию, которая будет отменять

изменения DOM-элементов, описанные в пункте «Неактивное состояние»

технического задания. \*/

//почему то в моем шаблоне notice\_\_form вместо ад форм

var noticeForm = document.querySelector('.notice\_\_form');

var formElements = noticeForm.querySelectorAll('.form\_\_element');

var activateMap = function () {

    map.classList.remove('map--faded');

    noticeForm.classList.remove('notice\_\_form--disabled');

    document.querySelector('.notice\_\_header').removeAttribute('disabled');

    for (var i = 0; i<formElements.length; i++) {

        formElements[i].disabled=false;

    }

}

var mapPin= document.querySelector('.map\_\_pin--main');

mapPin.addEventListener('mouseup', function () {

activateMap();

}

);

# VERSION 2

/\*Создайте массив, состоящий из 8 сгенерированных JS объектов, которые

будут описывать похожие объявления неподалёку \*/

// объявляю массив

var similarHotels = [];

//как забить массив заданными числовыми значениями

var arrayPush = function (min, max) {

    var newArray = [];

  for (var i = min; i<= max; i++) {

    newArray.push(i);

  }

    return newArray;

}

var numbersArray = arrayPush(1, 8);

//функция возвращающая случайное число из заданного диапазона

var renderRandomNumber = function (min, max) {

    var randomNumber = Math.floor(Math.random() \* (max - min + 1) + min);

    return randomNumber;

    }

//функция выбора случайного элемента из массива

var renderRandomArrayItem = function(array) {

    var i = renderRandomNumber(0, array.length-1);

    var randomItem = array[i];

    return randomItem;

}

// функция случайного перемещивания элементов массива

function shuffle(array) {

    var currentIndex = array.length, temporaryValue, randomIndex;

    while (0 !== currentIndex) {

      randomIndex = Math.floor(Math.random() \* currentIndex);

      currentIndex -= 1;

      temporaryValue = array[currentIndex];

      array[currentIndex] = array[randomIndex];

      array[randomIndex] = temporaryValue;

    }

    return array;

  }

//функция выдающая несколько случайных элементов из массива // случайным должно быть число окончания цикла. и до этого перемеживать надо

var pushRandomArrayItems = function (array) {

    shuffle(array);// перемешаем массив

    var randomArrayItems = '';

    for (var i = 0; i<= renderRandomNumber(0, array.length); i++) {

    randomArrayItems += array[i] +' ';

    }

    return randomArrayItems;

}

// все данные, которые будем использовать.

var TITLE\_DATA = ["Большая уютная квартира", "Маленькая неуютная квартира", "Огромный прекрасный дворец", "Маленький ужасный дворец", "Красивый гостевой домик", "Некрасивый негостеприимный домик", "Уютное бунгало далеко от моря", "Неуютное бунгало по колено в воде"];

var TYPE\_DATA = ['palace', 'flat', 'house', 'bungalo'];

var CHECKINOUT\_DATA = ['12:00', '13:00', '14:00'];

var FEATURES\_DATA = ["wifi", "dishwasher", "parking", "washer", "elevator", "conditioner"]; // надо сделать массив строк случайной длины из ниже предложенных

var PHOTOS\_DATA = ["http://o0.github.io/assets/images/tokyo/hotel1.jpg", "http://o0.github.io/assets/images/tokyo/hotel2.jpg", "http://o0.github.io/assets/images/tokyo/hotel3.jpg"];

//создадим функцию создания объектов с такими данными.

var generateHotelsArray = function () {

    var similarObject = {};

for (var i = 0; i < 8; i++) {

    similarObject = {

    author: {

        avatar: 'img/avatars/user' + '0' + numbersArray[i] + '.png'

    },

    offer: {

        title: shuffle(TITLE\_DATA)[i],

        address: renderRandomNumber(300, 900) + ', ' + renderRandomNumber(100, 500),

        price: renderRandomNumber(1000, 1000000),

        type: renderRandomArrayItem(TYPE\_DATA),

        rooms: renderRandomNumber(1, 5),

        guests: renderRandomNumber(1, 10), //????????????????????????

        checkin: renderRandomArrayItem(CHECKINOUT\_DATA),

        checkout: renderRandomArrayItem(CHECKINOUT\_DATA),

        features: pushRandomArrayItems(FEATURES\_DATA),

        description: '',

        photos: shuffle(PHOTOS\_DATA),

        location: {

            x: renderRandomNumber(300, 900),

            y: renderRandomNumber(130, 630)

        }

    }

    }

similarHotels.push(similarObject);

}

    return similarHotels;

}

similarHotels = generateHotelsArray();

console.log(similarHotels);

// 2 ЗАДАНИЕ

var map = document.querySelector('.map');

/\* map.classList.remove('map--faded'); \*/

//map\_\_pins - куда вставлять все ПИНы

var hotelsContainer = document.querySelector('.map\_\_pins');

//На основе данных, созданных в первом пункте, создайте DOM-элементы, соответствующие меткам на карте, и заполните их данными из массива.

//Итоговую разметку метки .map\_\_pin можно взять из шаблона .map\_\_card.

// элемент, КУДА будем копировать разметку из шаблона

var hotelPin = document.querySelector('.map\_\_pin');

// ЭТО ПИН

var pinTemplate = document.querySelector('template')

    .content

    .querySelector('.map\_\_pin');

// добавляю все это в функцию

var generatePinFromTemplate = function (hotel) {

    var hotelPinElement = pinTemplate.cloneNode(true);

    hotelPinElement.style.top = renderRandomNumber(100, 500) + 'px'; // ТАК ЗАДАЮТСЯ КООРДИНАТЫ!!!!!!!!!! задавать надо ПИНУ

    hotelPinElement.style.left = renderRandomNumber(300, 900) + 'px';

    var pinAvatar = hotelPinElement.querySelector('img');

    pinAvatar.src = hotel.author.avatar;

    pinAvatar.alt = hotel.offer.description;

    return hotelPinElement;

}

//Отрисуйте сгенерированные DOM-элементы в блок .map\_\_pins. Для вставки элементов используйте DocumentFragment.

// ФУНКЦИЯ ДОБАВЛЕНИЯ В ВЕРСТКУ ЧЕРЕЗ ДОКУМЕНТ ФРАГМЕНТ

    var insertMapPins = function () {

        var fragment = document.createDocumentFragment();

    for (var i = 0; i < similarHotels.length; i++) {

        var newPin = generatePinFromTemplate(similarHotels[i]);

        newPin.classList.add('pin--'+(i+1));

        fragment.appendChild(newPin);

    }

    hotelsContainer.appendChild(fragment);

    };

    insertMapPins(); // отрисовываю метки

// функции скрытия и показа отрисованных меток

var hideMapPins = function () {

    var mapPins = document.querySelectorAll('.map\_\_pin');

    for (var i = 1; i <= similarHotels.length; i++ ) {

        mapPins[i].classList.add('hidden');

    }

}

hideMapPins();

var showMapPins = function () {

    var mapPins = document.querySelectorAll('.map\_\_pin');

    for (var i = 1; i <= similarHotels.length; i++ ) {

        mapPins[i].classList.remove('hidden');

    }

}

// ОБЪЯВЛЕНИЯ

/\*.На основе ПЕРВОГО по порядку элемента ИЗ СГЕНЕРИРОВАННОГО МАССИВА и шаблона .map\_\_card создайте DOM-элемент объявления, заполните его

данными из объекта и вставьте полученный DOM-элемент

в блок .map перед блоком.map\_\_filters-container\*/

//Можно задать переменную для .map\_\_filters-container перед которым надо вставить элементы. И затем вместо аппенд использовать бифор

// map - КУДА ВСТАВЛЯТЬ

var mapFilter= document.querySelector('.map\_\_filters-container');

//ШАБЛОН, КОТОРЫЙ БУДЕТ КОПИРОВАТЬ -- ЭТО ОБЪЯВЛЕНИЕ

 var hotelTemplate = document.querySelector('template')

    .content

    .querySelector('.map\_\_card');

// по порядку задаю значения, но не добавляю никуда

var hotelElement = hotelTemplate.cloneNode(true);

var title = hotelElement.querySelector('h3');

title.classList.add('popup\_\_title');

title.textContent = similarHotels[0].offer.title;

var address = hotelElement.querySelector('p').querySelector('small');

address.classList.add('popup\_\_text--address');

address.textContent = similarHotels[0].offer.address;

var price = hotelElement.querySelector('.popup\_\_price');

price.classList.add('popup\_\_text--price');

price.textContent = similarHotels[0].offer.price + ' ₽/ночь';

var type = hotelElement.querySelector('h4');

type.classList.add('popup\_\_type');

var typeName = similarHotels[0].offer.type;

if(typeName == 'flat'){

    type.textContent ='Квартира';

}

if(typeName == 'bungalo'){

    type.textContent ='Бунгало';

}

if(typeName == 'palace') {

    type.textContent ='Дворец';

}

;

var capacity = hotelElement.querySelector('p:nth-of-type(3)');

capacity.classList.add('popup\_\_text--capacity');

capacity.textContent = similarHotels[0].offer.rooms + ' комната(ы) для ' + similarHotels[0].offer.guests + ' гост(я)ей';

var time = hotelElement.querySelector('p:nth-of-type(4)');

time.classList.add('popup\_\_text--time');

time.textContent = 'Заезд после ' + similarHotels[0].offer.checkin + ', выезд до ' + similarHotels[0].offer.checkout ;

/\* в шаблоне уже есть список с классами. сделать проверку, если в массиве нет значения Н, то удалить класс Н

\*/

var popupFeatures = hotelElement.querySelector('.popup\_\_features'); // нашел УЛ

var featureItems = similarHotels[0].offer.features.split(' '); //выводится массив со случайными значениями

console.log(featureItems);

//сматчить каждый элемент из списка ул

var wifiFeature = hotelElement.querySelector('.popup\_\_features').querySelector('li:nth-child(1)');

var dishwasherFeature = hotelElement.querySelector('.popup\_\_features').querySelector('li:nth-child(2)');

var parkingFeature = hotelElement.querySelector('.popup\_\_features').querySelector('li:nth-child(3)');

var washerFeature = hotelElement.querySelector('.popup\_\_features').querySelector('li:nth-child(4)');

var elevatorFeature = hotelElement.querySelector('.popup\_\_features').querySelector('li:nth-child(5)');

var conditionerFeature = hotelElement.querySelector('.popup\_\_features').querySelector('li:nth-child(6)');

console.log(wifiFeature);

if (featureItems.indexOf('wifi') == -1) {

    popupFeatures.removeChild(wifiFeature);

}

if (featureItems.indexOf('dishwasher') == -1) {

    popupFeatures.removeChild(dishwasherFeature);

}

if (featureItems.indexOf('parking') == -1) {

    popupFeatures.removeChild(parkingFeature);

}

if (featureItems.indexOf('washer') == -1) {

    popupFeatures.removeChild(washerFeature);

}

if (featureItems.indexOf('elevator') == -1) {

    popupFeatures.removeChild(elevatorFeature);

}

if (featureItems.indexOf('conditioner') == -1) {

    popupFeatures.removeChild(conditionerFeature);

}

var popupDescription = hotelElement.querySelector('p:nth-of-type(5)');

popupDescription.classList.add('.popup\_\_description');

popupDescription.textContent = similarHotels[0].offer.description;

var popupPhotos = hotelElement.querySelector('.popup\_\_pictures');

popupPhotos.classList.add('.popup\_\_photos');

popupPhotos.querySelector('li:nth-child(1)').querySelector('img').src = similarHotels[0].offer.photos[0];

popupPhotos.querySelector('li:nth-child(1)').querySelector('img').classList.add('popupPhoto');

popupPhotos.querySelector('li:nth-child(2)').querySelector('img').src = similarHotels[0].offer.photos[1];

popupPhotos.querySelector('li:nth-child(2)').querySelector('img').classList.add('popupPhoto');

popupPhotos.querySelector('li:nth-child(3)').querySelector('img').src = similarHotels[0].offer.photos[2];

popupPhotos.querySelector('li:nth-child(3)').querySelector('img').classList.add('popupPhoto');

popupPhotos.querySelector('.popupPhoto').naturalWidth = 10;

var popupAvatar = hotelElement.querySelector('img');

popupAvatar.src = similarHotels[0].author.avatar;

//выводим окно когда кликнут по МЕТКЕ СКОПИРУЮ ЭТО В СОБЫТИЕ КЛИКА

/\* map.insertBefore(hotelElement, mapFilter); \*/

//ЗАДАНИЕ 4

//АКТИВАЦИЯ страницы. по клику должна активироваться карта, появится метки, поле формы стать активным, адрес должен заполнится. и надо потом добавить проверку формы.

/\* Первое действие, которое нужно выполнить, перед тем, как приступить к этому

заданию, вернуть страницу в исходное состояние \*/

//добавить класс .map--faded. я просто в комментарии убрал в начале кода

console.log(map);

/\* поля формы должны быть неактивны в исходном состоянии. \*/

//добавил в верстке. удалять все равно потом через ДжиЭс

/\* 1. Активация страницы

Для этого нужно добавить обработчик

события mouseup на элемент .map\_\_pin--main.

Обработчик события mouseup должен вызывать функцию, которая будет отменять

изменения DOM-элементов, описанные в пункте «Неактивное состояние»

технического задания. \*/

//почему то в моем шаблоне notice\_\_form вместо ад форм

var noticeForm = document.querySelector('.notice\_\_form');

var formElements = noticeForm.querySelectorAll('.form\_\_element');

var activateMap = function () {

    map.classList.remove('map--faded');

    noticeForm.classList.remove('notice\_\_form--disabled');

    document.querySelector('.notice\_\_header').removeAttribute('disabled');

    for (var i = 0; i<formElements.length; i++) {

        formElements[i].disabled=false;

    }

}

var mapPinMain= document.querySelector('.map\_\_pin--main');

mapPinMain.addEventListener('mouseup', function () {

activateMap();

showMapPins();

}

);

// по клику на метки показывать страницу описания ДЕЛЕГИРОВАНИЕ

// отлавливать клики по контейнеру

console.log(hotelsContainer);

hotelsContainer.addEventListener('click', function (evt) {

    var target = evt.target.closest('button'); //так клик показывает кнопку, даже если клик по изображению внутри кнопки

    if (target.classList.contains('pin--1')) {

    }

})