**Лабораторная работа №7**

**Работа с DOM**

Сдать до 25.11

**Цель:** Разобраться с DOM API. Понять связь между исполняемым скриптом и DOM-деревом. Отображать данные с помощью JavaScript и DOM API.

**Задание**

**1. Отобразить данные с помощью JavaScript и DOM API**

Требуется реализовать функции, которые будут манипулировать DOM-деревом в соответствии с бизнес-логикой приложения:

1. Отобразить информацию на HTML-странице.

2. Добавить информацию на HTML-страницу.

3. Удалить информацию с HTML-страницы

4. Отредактировать информацию на HTML-странице

5. Отобразить элементы, зависящие от наличия текущего пользователя или его действий: кнопки для редактирования, удаления, добавления, поиска информации. (В коде сделайте состояние пользователя легко конфигурируемым, например, в начале файла константа var user = ‘Ivan Ivanov’ или var user = null, если его нет) на HTML-странице

6. Использовать фильтр (выбор конкретной информации).

**2. Реализовать класс View**

Аналогично, как и в предыдущем задании, необходимо реализовать класс для работы с отображением страницы.

Класс не обязательно должен быть один, вы можете разделить логику отображения разных частей страницы, если они независимы (например, *HeaderView, TapeView, EditView*).

**3. Реализовать функции для демонстрации работы бизнес-логики приложения**

Требуется в глобальной области видимости реализовать функции, которые продемонстрируют работу бизнес-логики приложения, манипулируя DOMом.

Например, метод addObj(ObjInf: Object), должен вызвать соответствующий метод из класса для работы с моделью данных, и, если новая информация была добавлена успешно, вызвать метод для отображения изменений в DOMе (т.е. вызвать метод, разработанный в рамках пункта 1).

*Глобальные функции следует обернуть в модуль, чтоб вызывать потом из консоли.*

**Итого, что необходимо сделать в рамках данного задания:**

После выполнения данного задания ваша информация (ObjInf) должна отображаться с помощью JavaScript. В исходном index.html информации об объектах( ObjInf ) быть не должно.

С помощью глобальных функций (пункт 3) можно добавлять новую информацию (ObjInf)согласно индивидуальному заданию), изменять и удалять существующие.

Вызов этих функций изменяет модель списка вашей информации (ObjInf) в JavaScript (написанную в задании 5), а также визуальную информацию(кнопки и т.д), отображаемую в браузере.

То есть, обработчиков событий еще нет. Действия пользователя имитируются вызовами глобальных функций **из консоли.**

*Пример:*

*1. Открываем index.html в браузере.*

*2. Подключенный script.js отображает информацию из массива ObjInf (подпункт 1.1). Отображает имя текущего пользователя (подпункт 1.5). Заполняет опции фильтра (1.6).*

*3. Имитируем действия пользователя вызовами из консоли функций из подпункта 3:*

*a. Вызов addObj({description: ‘description’, ...}) - добавляет новую инф. в массив ObjInf и отображает новый Obj в списке в браузере*

*(в нужную позицию в соответствии с условиями сортировки).*

*b. Вызов removeObj(id) - удаляет Obj из массива ObjInf и удаляет Obj из списка в браузере (если он отображается).*

*c. Вызов editObj(id, {}) - изменяет Obj в массиве и изменяет соответствующий Obj в списке в браузере (если он отображается).*