Лабораторная работа №3 (часть1)

“Работа с DOM. События”

**Вариант** 4

**Задание 1.**

Функция должна создать элемент с тегом DIV, поместить в него текстовый узел и вернуть получившийся элемент

@param {string} text - текст, который необходимо поместить в div

@return {Element}

*function createDivWithText(text) {*

*}*

**Задание 2.**

Функция должна отслеживать добавление и удаление элементов внутри элемента where. Как только в where добавляются или удаляются элемента, необходимо сообщать об этом при помощи вызова функции fn со специальным аргументом. В качестве аргумента должен быть передан объек с двумя свойствами:

- type: типа события (insert или remove)

- nodes: массив из удаленных или добавленных элементов (а зависимости от события)

Отслеживание должно работать вне зависимости от глубины создаваемых/удаляемых элементов

Рекомендуется использовать MutationObserver

@param {Element} where - где отслеживать

@param {function(info: {type: string, nodes: Array<Element>})} fn - функция, которую необходимо вызвать

@example

если в where или в одного из его детей добавляется элемент div то fn должна быть вызвана с аргументов:

{

type: 'insert',

nodes: [div]

}

если из where или из одного из его детей удаляется элемент div то fn должна быть вызвана с аргументов:

{

type: 'remove',

nodes: [div]

}

*function observeChildNodes(where, fn) {*

*}*

**Задание 3**

Функция должна добавлять обработчик fn события eventName к элементу target

@param {string} eventName - имя события, на которое нужно добавить обработчик

@param {Element} target - элемент, на который нужно добавить обработчик

@param {function} fn - обработчик

*function addListener(eventName, target, fn) {*

*}*