Лабораторная работа №8. Объекты. Глубокое копирование.

**Задание**: изучит теорию

*Тема 2: Структуры данных. Вопрос 6:**Объекты.*

**Задача.**

1. Выполнить глубокое копирование всех объектов и массивов. Использовать spread оператор.
2. Обратитесь к копии объекта user5 и измените значение свойства group на 12, а оценку по программированию измените на 10.
3. Обратитесь к копии объекта user6 и измените имя преподавателя.
4. Обратитесь к копии объекта user7 и измените количество страниц на 3 для статьи “About CSS” преподавателя Petr Ivanov.
5. Замените все сообщения в объекте store на “Hello”.

1.

let  ***user*** = {  
 name: 'Masha',  
 age: 21  
};

2.

let ***numbers*** = [1, 2, 3];

3.

let ***user1*** = {  
 name: 'Masha',  
 age: 23,  
 location: {  
 city: 'Minsk',  
 country: 'Belarus'  
 }  
};

4.

let ***user2*** = {  
 name: 'Masha',  
 age: 28,  
 skills: ["HTML", "CSS", "JavaScript", "React"]  
};

5.

const array = [  
 {id: 1, name: 'Vasya', group: 10},   
 {id: 2, name: 'Ivan', group: 11},  
 {id: 3, name: 'Masha', group: 12},  
 {id: 4, name: 'Petya', group: 10},  
 {id: 5, name: 'Kira', group: 11},  
]

6.

let ***user4*** = {  
 name: 'Masha'',  
 age: 19,  
 studies: {  
 university: 'BSTU',  
 speciality: 'designer',  
 year: 2020,  
 exams: {  
 maths: true,  
 programming: false  
 }  
 }  
};

7.

let ***user5*** = {  
 name: 'Masha'',  
 age: 22,  
 studies: {  
 university: 'BSTU',  
 speciality: 'designer',  
 year: 2020,  
 department: {  
 faculty: 'FIT',  
 group: 10,  
 }  
 exams: [  
 { maths: true, mark: 8},  
 { programming: true, mark: 4},  
 ]  
 }  
};

8.

let ***user6*** = {  
 name: 'Masha'',  
 age: 21,  
 studies: {  
 university: 'BSTU',  
 speciality: 'designer',  
 year: 2020,  
 department: {  
 faculty: 'FIT',  
 group: 10,  
 }  
 exams: [  
 {

maths: true,

mark: 8,

professor: {

name: 'Ivan Ivanov ',

degree: 'PhD'

}

},  
 {

programming: true,

mark: 10,

professor: {

name: 'Petr Petrov',

degree: 'PhD'

}

},  
 ]  
 }  
};

9.

let ***user7*** = {  
 name: 'Masha'',  
 age: 20,  
 studies: {  
 university: 'BSTU',  
 speciality: 'designer',  
 year: 2020,  
 department: {  
 faculty: 'FIT',  
 group: 10,  
 }  
 exams: [  
 {

maths: true,

mark: 8,

professor: {

name: 'Ivan Petrov',

degree: 'PhD',

articles: [  
 {title: "About HTML", pagesNumber: 3},  
 {title: "About CSS", pagesNumber: 5},  
 {title: "About JavaScript", pagesNumber: 1},  
 ]

}

},  
 {

programming: true,

mark: 10,

professor: {

name: 'Petr Ivanov',

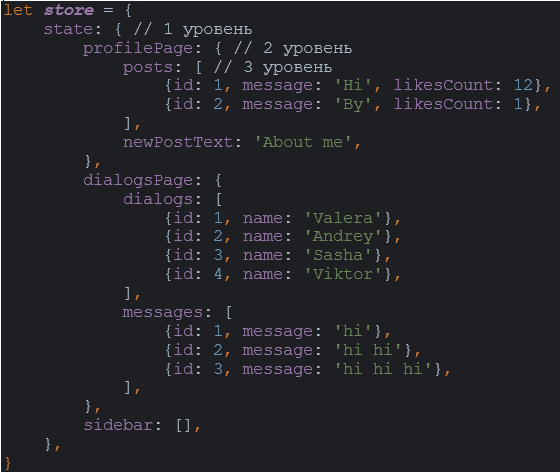
degree: 'PhD',

articles: [  
 {title: "About HTML", pagesNumber: 3},  
 {title: "About CSS", pagesNumber: 5},  
 {title: "About JavaScript", pagesNumber: 1},  
 ]

}

},  
 ]  
 }  
};

10.



1. **Массивы (Arrays):** Массивы в JavaScript представляют собой упорядоченные списки элементов. Элементы могут быть любого типа данных, в том числе другие массивы.

javascriptCopy code

let fruits = ['apple', 'orange', 'banana'];

Операции с массивами включают доступ к элементам по индексу, добавление новых элементов, удаление элементов и многое другое.

1. **Объекты (Objects):** Объекты в JavaScript используются для создания коллекций пар ключ-значение. Ключи (или свойства) являются строками или символами, а значения могут быть любого типа данных.

javascriptCopy code

let person = { name: 'John', age: 30, isStudent: false };

Доступ к свойствам объекта осуществляется через точку (**person.name**) или квадратные скобки (**person['name']**).

1. **Стеки (Stacks):** Стек — это коллекция элементов, управляемая по принципу "последний вошел, первый вышел" (Last-In-First-Out, LIFO). Операции включают добавление элемента на вершину стека и удаление элемента с вершины стека.

javascriptCopy code

let stack = []; stack.push(1); stack.push(2); stack.pop(); // Возвращает 2

1. **Очереди (Queues):** Очередь — это коллекция элементов, управляемая по принципу "первый вошел, первый вышел" (First-In-First-Out, FIFO). Операции включают добавление элемента в конец очереди и удаление элемента с начала очереди.

javascriptCopy code

let queue = []; queue.push(1); queue.push(2); queue.shift(); // Возвращает 1

1. **Множества (Sets):** Множество в JavaScript представляет собой коллекцию уникальных элементов. Операции включают добавление элемента, удаление элемента и проверку наличия элемента в множестве.

javascriptCopy code

let set = new Set(); set.add(1); set.add(2); set.has(1); // Возвращает true

1. **Карты (Maps):** Карты представляют собой коллекцию пар ключ-значение, где ключи могут быть любого типа данных. Карты предоставляют методы для добавления, удаления и поиска элементов.

javascriptCopy code

let map = new Map(); map.set('name', 'Alice'); map.get('name'); // Возвращает 'Alice'

1. **Связанные списки (Linked Lists):** Связанный список состоит из узлов, где каждый узел содержит значение и ссылку на следующий узел. Операции включают добавление элемента, удаление элемента и перебор элементов.

javascriptCopy code

class Node { constructor(value) { this.value = value; this.next = null; } } let linkedList = new Node(1); linkedList.next = new Node(2);