**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ**

***Институт Принтмедиа и информационных технологий Кафедра Информатики и информационных технологий***

**направление подготовки 09.03.02 «Информационные системы и технологии»**

# **ЛАБОРАТОРНАЯ РАБОТА 5**

**Дисциплина:** Технологии кроссплатформенного программирования

**Тема:** Объекты в JS

**Выполнил: студент группы 211-724**

**Беляев В.Д.**

(Фамилия И.О.)

**Дата, подпись**  17.03.2023 \_

(Дата) (Подпись)

## **Проверил:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ***\_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание)(Оценка)

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

## **Замечания:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Москва**

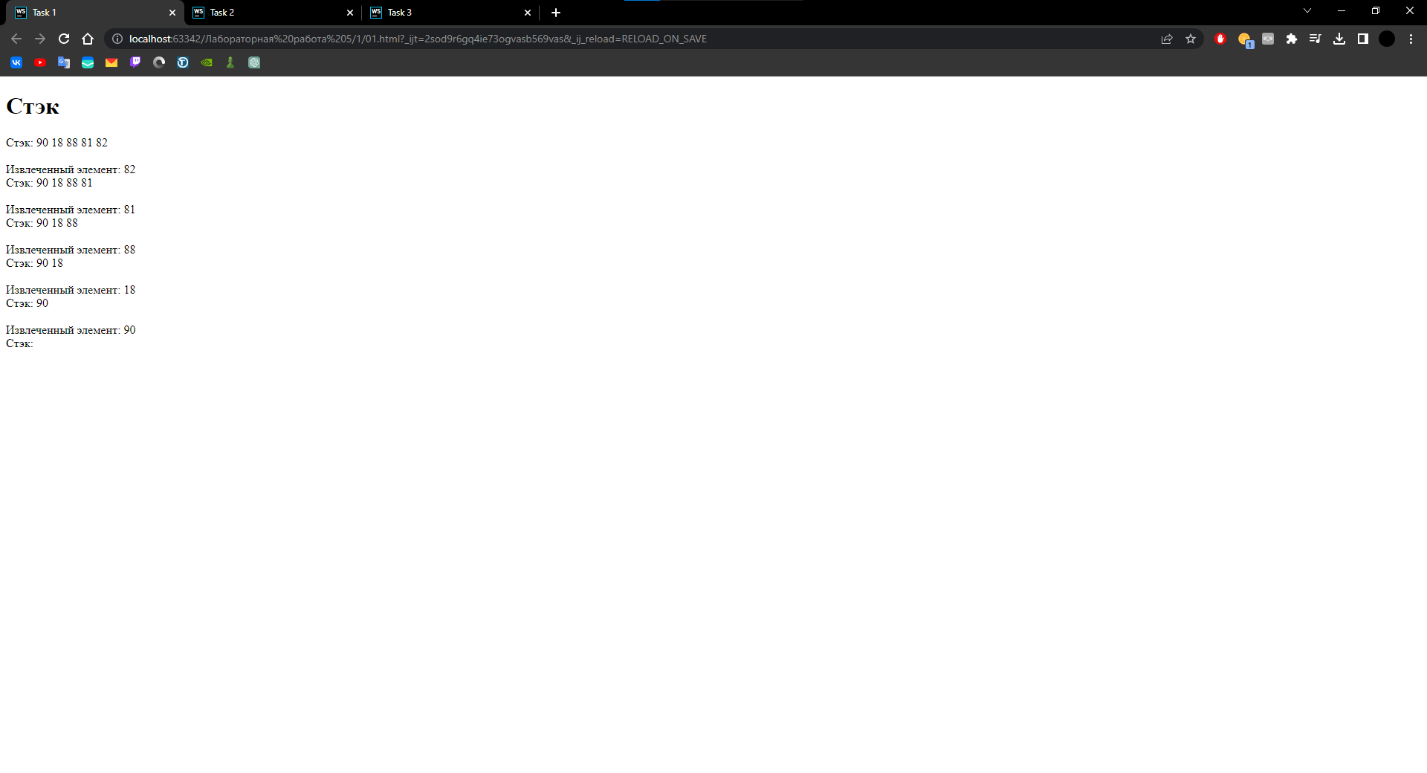
**2023**

**Файл 01.html**

<!DOCTYPE html>  
<html lang = "en">  
<head>  
 <meta charset = "UTF-8">  
 <title>Task 1</title>  
 <script src = '1.js'></script>  
</head>  
<body>  
  
</body>  
</html>

**Файл 1.js**

class Stack {  
 constructor() {  
 this.arr = []  
 }  
  
 insert(item) {  
 this.arr.push(item)  
 }  
  
 extract() {  
 return this.arr.pop()  
 }  
  
 getSize() {  
 return this.arr.length  
 }  
  
 getItems() {  
 return this.arr  
 }  
}  
  
let getRandomInt = function (start, end) {  
 return start + ***Math***.ceil(***Math***.random() \* (***Math***.max(start, end) - ***Math***.min(start, end)))  
}  
  
  
// Стэк  
***document***.writeln(`<h1>Стэк</h1>`)  
let ***stack*** = new Stack()  
  
for (let i = 0; i < 5; ++i)  
 ***stack***.insert(getRandomInt(0, 100))  
  
***document***.writeln(`Стэк: ${***stack***.getItems().map(item => item.toString()).join(' ')}<br>`)  
while (***stack***.getSize()) {  
 ***document***.writeln(`<br>Извлеченный элемент: ${***stack***.extract()}<br>`)  
 ***document***.writeln(`Стэк: ${***stack***.getItems().map(item => item.toString()).join(' ')}<br>`)  
}

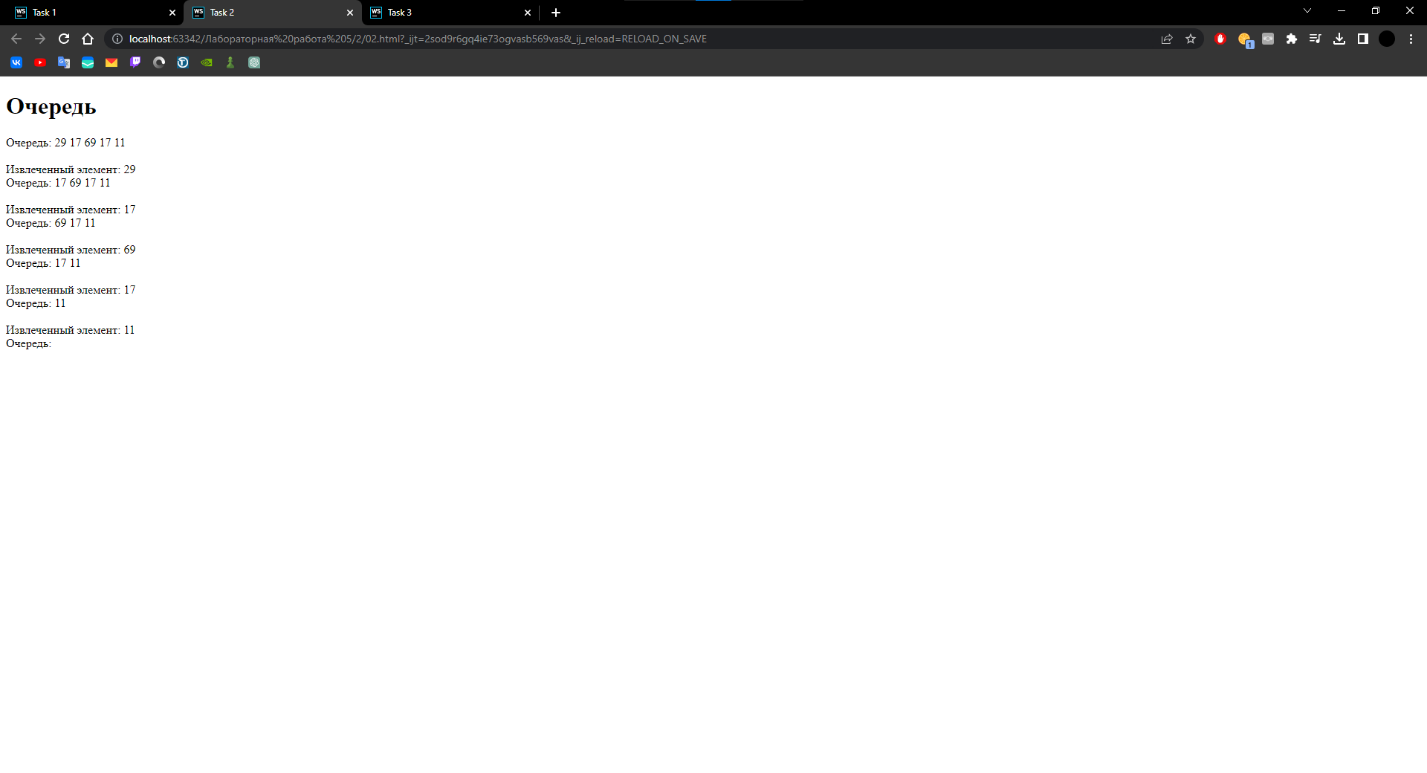
****

**Файл 02.html**

<!DOCTYPE html>  
<html lang = "en">  
<head>  
 <meta charset = "UTF-8">  
 <title>Task 2</title>  
 <script src = '2.js'></script>  
</head>  
<body>  
  
</body>  
</html>

**Файл 2.js**

class Queue {  
 constructor() {  
 this.arr = []  
 }  
  
 insert(item) {  
 this.arr.push(item)  
 }  
  
 extract() {  
 let result = this.arr[0]  
 this.arr.shift()  
 return result  
 }  
  
 getSize() {  
 return this.arr.length  
 }  
  
 getItems() {  
 return this.arr  
 }  
}  
  
let getRandomInt = function (start, end) {  
 return start + ***Math***.ceil(***Math***.random() \* (***Math***.max(start, end) - ***Math***.min(start, end)))  
}  
  
// Очередь  
***document***.writeln(`<h1>Очередь</h1>`)  
let ***queue*** = new Queue()  
  
for (let i = 0; i < 5; ++i)  
 ***queue***.insert(getRandomInt(0, 100))  
  
***document***.writeln(`Очередь: ${***queue***.getItems().map(item => item.toString()).join(' ')}<br>`)  
while (***queue***.getSize()) {  
 ***document***.writeln(`<br>Извлеченный элемент: ${***queue***.extract()}<br>`)  
 ***document***.writeln(`Очередь: ${***queue***.getItems().map(item => item.toString()).join(' ')}<br>`)  
}

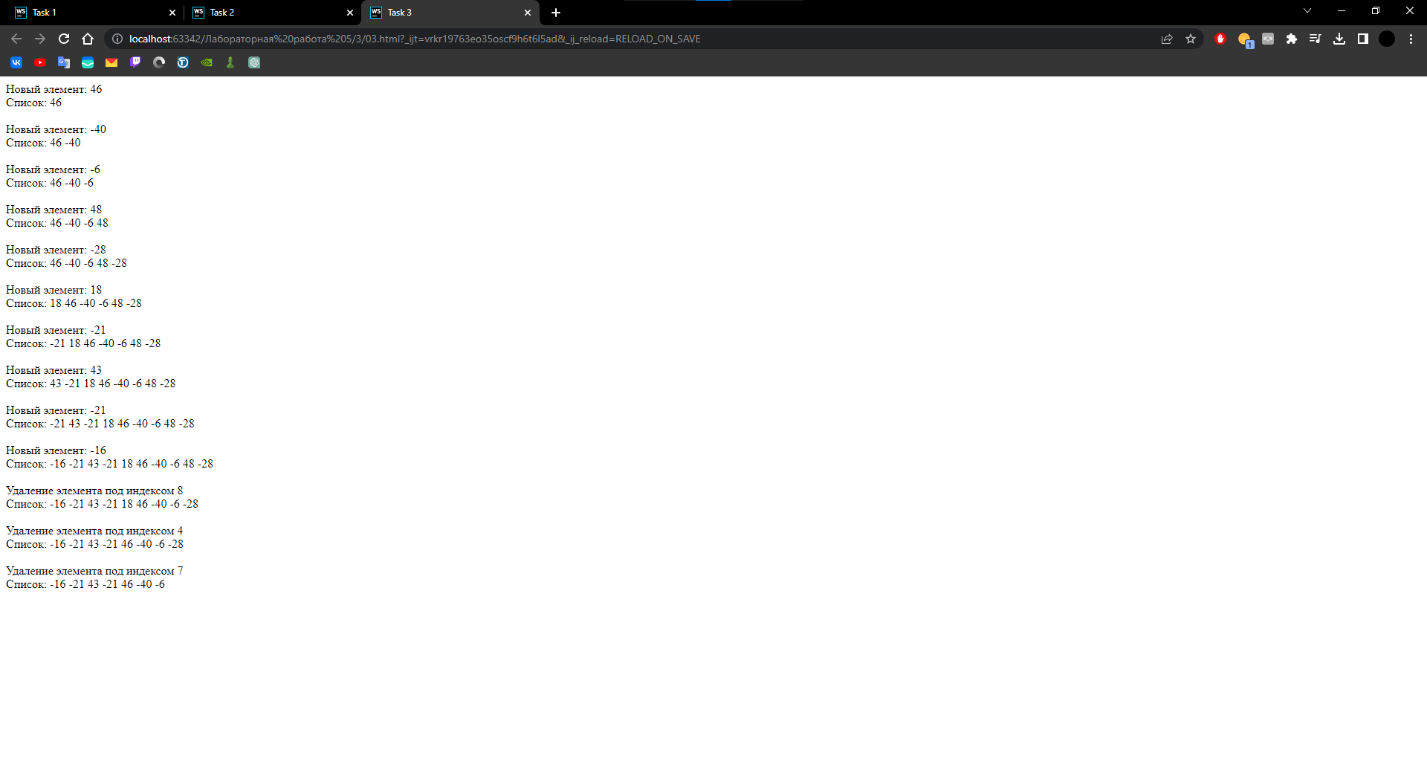
****

**Файл 03.html**

<!DOCTYPE html>  
<html lang = "en">  
<head>  
 <meta charset = "UTF-8">  
 <title>Task 3</title>  
 <script src = '3.js'></script>  
</head>  
<body>  
  
</body>  
</html>

**Файл 3.js**

class Node {  
 constructor(data) {  
 this.prev = null  
 this.data = data  
 this.next = null  
 }  
}  
  
class LinkedList {  
 constructor() {  
 this.head = null  
 this.tail = null  
 }  
  
 pushBack(data) {  
 let newNode = new Node(data)  
  
 if (!this.head) {  
 this.head = newNode  
 this.tail = this.head  
 } else {  
 newNode.prev = this.tail  
 this.tail.next = newNode  
 this.tail = newNode  
 }  
  
 return this.toArray()  
 }  
  
 pushFront(data) {  
 let newNode = new Node(data)  
  
 if (!this.head) {  
 this.head = newNode  
 this.tail = this.head  
 } else {  
 newNode.next = this.head  
 this.head.prev = newNode  
 this.head = newNode  
 }  
  
 return this.toArray()  
 }  
  
 remove(index) {  
 let current = this.head;  
  
 for (let i = 0; i < index; ++i)  
 current = current.next  
  
 if (current === this.head && current === this.tail) {  
 this.head = null;  
 this.tail = null;  
 } else if (current === this.head) {  
 this.head = this.head.next;  
 this.head.prev = null;  
 } else if (current === this.tail) {  
 this.tail = this.tail.prev;  
 this.tail.next = null;  
 } else {  
 current.prev.next = current.next;  
 current.next.prev = current.prev;  
 }  
  
 return this.toArray()  
 }  
  
 toArray() {  
 let node = this.head  
 let result = []  
  
 while (node) {  
 result.push(node.data)  
 node = node.next  
 }  
  
 return result  
 }  
}  
  
let getRandomInt = function (start, end) {  
 return start + ***Math***.ceil(***Math***.random() \* (***Math***.max(start, end) - ***Math***.min(start, end)))  
}  
  
// Создание списка  
let ***list*** = new LinkedList()  
  
// Добавление элементов в список: сначала в конец, потом в начало  
let ***actions*** = [(number) => ***list***.pushBack(number), (number) => ***list***.pushFront(number)]  
***actions***.forEach(action => {  
 for (let i = 0; i < 5; ++i) {  
 let value = getRandomInt(-50, 50)  
 action(value)  
  
 ***document***.writeln(`Новый элемент: ${value}<br>`)  
  
 ***document***.writeln(`Список: `)  
 ***list***.toArray().forEach(item => {  
 ***document***.writeln(`${item} `)  
 })  
  
 ***document***.writeln(`<br><br>`)  
 }  
})  
  
// Удаление элементов из списка  
for (let i = 0; i < 3; ++i) {  
 let indexToRemove = getRandomInt(0, ***list***.toArray().length - 1)  
  
 ***document***.writeln(`Удаление элемента под индексом ${indexToRemove}<br>`)  
  
 ***document***.writeln(`Список: `)  
 ***list***.remove(indexToRemove).forEach(item => {  
 ***document***.writeln(`${item} `)  
 })  
  
 ***document***.writeln(`<br><br>`)  
}

****