**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(МИИТ)**

|  |
| --- |
| **ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Кафедра «Управление и защита информации»

Лабораторная работа №7

**«Жизненный цикл Vue.js»**

по дисциплине

**«Web программирование»**

|  |  |
| --- | --- |
|  | Выполнил: студент группы ТКИ-541  Куминов В. П.  Проверил: к.т.н., доц.  Сафронов А. И. |
|  |  |

**Москва – 2023 г.**

**1. Цель:**

Продемонстрировать понимание и способность применять жизненный цикл фреймворка Vue.js.3.x.

**2. Формулировка задачи:**

Продумать схему тестирования и демонстрации работы всех витков / хуков / методов жизненного цикла фреймворка Vue.js в формате одностраничного веб-приложения. Реализовать схему тестирования строго под Vue.js 3.x (Приложение 1). Русифицировать схему жизненного цикла Vue.js 3.x

Близкая реализация приложения для тестирования под Vue.js 2.x приведена в Приложении 2.

Привести информацию об используемом браузере по аналогии с иллюстрациями, представленными в Приложении 3.

Составить таблицу соответствия переменных и методов, используемых в web-приложении. Структура таблицы соответствия представлена в Приложение 4.

Заполнить спецификацию оборудования, используемого для выполнения работы Приложение 5.

Не использовать: jQuerry, «чистый» JavaScript, директиву v-for (Приложение 2).

Реализовать: локально, не прибегая к инструментарию Node.js и npm (Node Package Manager) – раскрыть данный вопрос в пункте 5 отчёта.

**3. Спецификация оборудования, на котором выполнялась работа.**

При выполнении работы использовался браузер Edge (Рисунок 1)

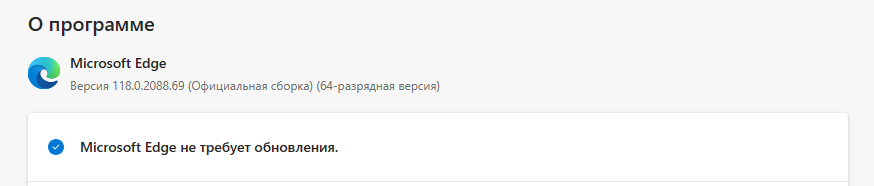
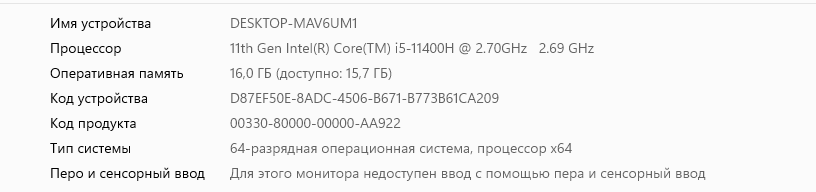
****

Рисунок 1. Браузер Microsoft Edge.

Характеристики устройства и Windows представлены на рисунке 2.

****

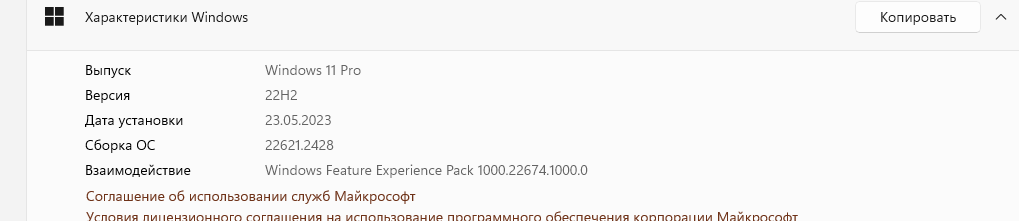
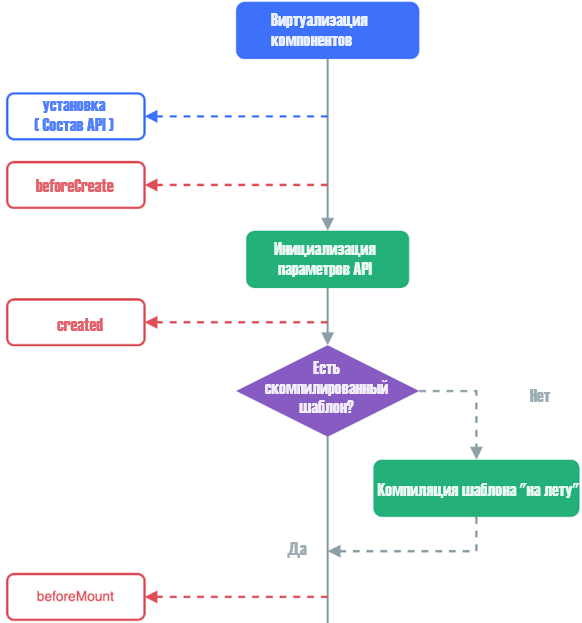
****

Рисунок 2 Спецификация оборудования

Таблица 1 – Спецификация оборудования.

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Характеристика** | **Сведения** |
| **1** | Тип устройства | Ноутбук |
| **2** | Модель | ASUS TUF Gaming F17 |
| **3** | Операционная система | *Microsoft Windows 11* |
| **4** | Процессор | *Intel Core i5* |
| **5** | Оперативная память | 16 Гб |
| **6** | Объём жёсткого диска | 512 Гб |

**4. Русифицированние схемы жизненного цикла *Vue.js*.**

****

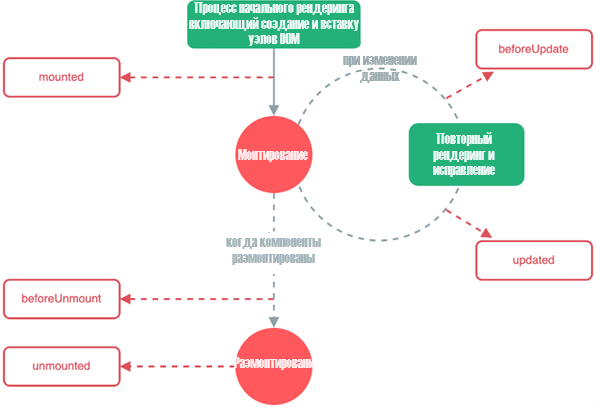
****

Рисунок 3 Жизненный цикл Vue.js

**5. Русифицированные схемы жизненного цикла *Vue.js*.**

Построим сеть Петри для приложения:

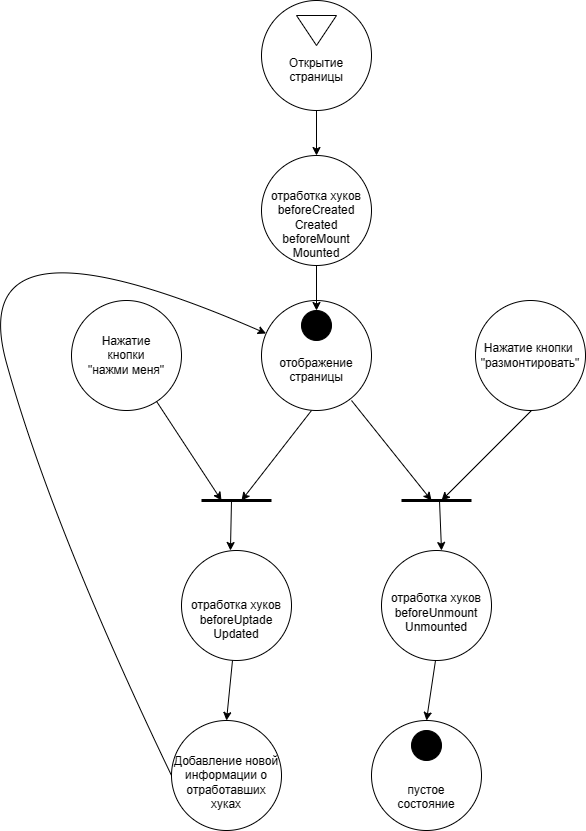


Рисунок 4 Сеть Петри

**Код программы**

<!DOCTYPE html>

<html>

<head>

<script src="https://unpkg.com/vue@3"></script>

<style>

#log {

border: 2px solid #000;

padding: 10px;

margin-top: 10px;

max-height: 200px;

overflow-y: auto;

}

.log-message {

margin-bottom: 10px;

}

</style>

</head>

<body>

<div id="app">

{{ message }}

<button v-on:click="changeMessage">Нажми меня</button>

<button id="unmount-button">Размонтировать</button>

<div id="log"></div>

</div>

<script>

const App = {

data() {

return {

message: 'Мы рождаемся!',

messages: ['Взрослеем!', 'Стареем!', 'Умираем!'],

index: 0,

logs: []

}

},

methods: {

changeMessage() {

this.index = (this.index + 1) % this.messages.length;

this.message = this.messages[this.index];

},

log(message) {

console.log(message);

const logDiv = document.getElementById('log');

if (logDiv) {

const newLogMessage = document.createElement('div');

newLogMessage.textContent = message;

newLogMessage.className = 'log-message';

logDiv.appendChild(newLogMessage);

}

}

},

mounted() {

this.log('created');

this.log('beforeMount');

this.log('mounted');

},

beforeUpdate() {

this.log('beforeUpdate');

},

updated() {

this.log('updated');

},

beforeUnmount() {

this.log('beforeUnmount');

},

unmounted() {

this.log('unmounted');

}

}

const appInstance = Vue.createApp(App);

const mountedApp = appInstance.mount('#app');

document.getElementById('unmount-button').addEventListener('click', function() {

appInstance.unmount();

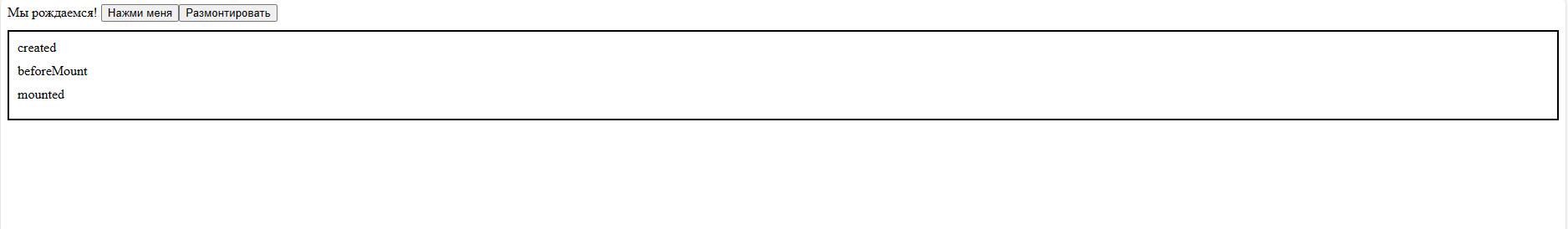
});

</script>

</body>

</html>

Результат работы кода можно видеть на рисунках 5-6



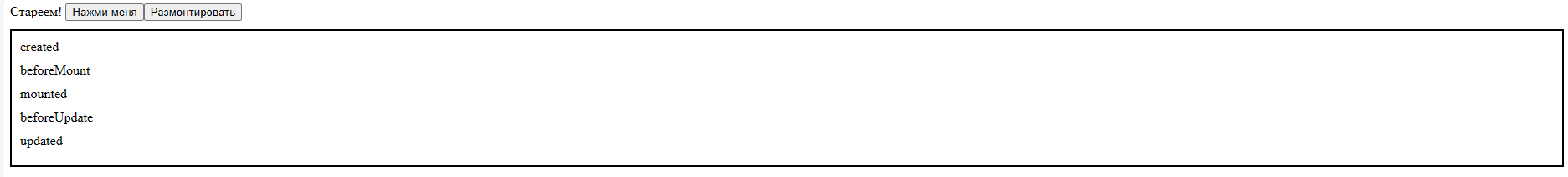
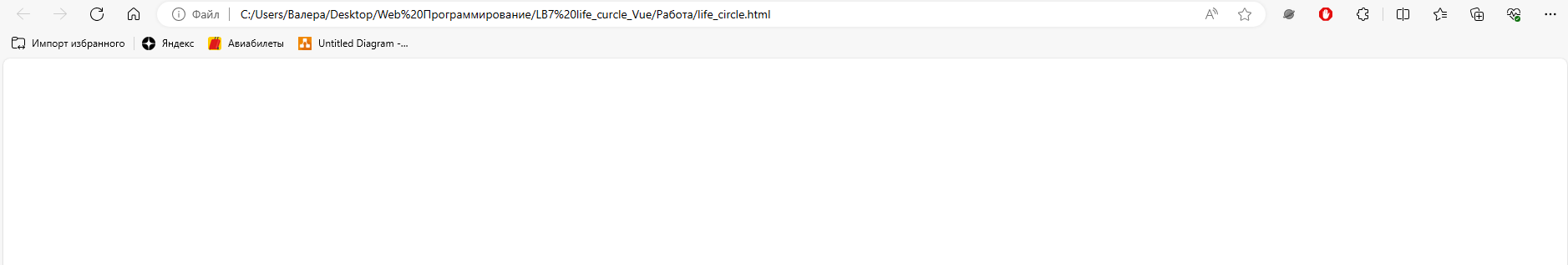


Рисунок 5 Нажатие кнопки «Нажми меня»



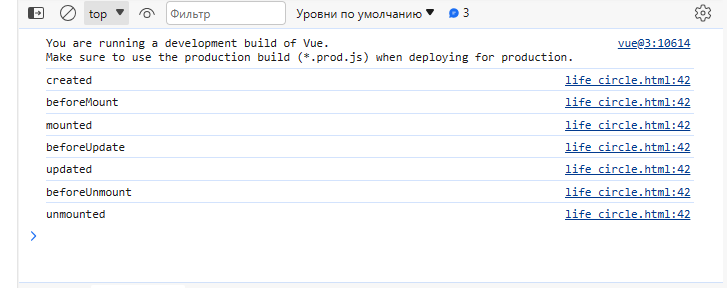


Рисунок 6 Нажатие кнопки «Размонтировать»

**6. Вывод**

Было создано приложение на Vue.js, которое отображает различные сообщения при нажатии кнопки. Приложение также отслеживает и отображает события жизненного цикла компонента, такие как монтирование, обновление и размонтирование. Для демонстрации процесса размонтирования добавлена функция, которая позволяет пользователю вручную размонтировать компонент. Это приложение служит наглядным примером того, как работают жизненные циклы в Vue.js.