|  |  |
| --- | --- |
| **Российский университет транспорта (МИИТ)**  **Институт транспортной техники и систем управления**  **Кафедра «Управление и защита информации»** | |
| **Отчет**  **по практическому заданию**  **по теме «Разработка с использованием Vue.js. Задание 6»**  **по дисциплине**  **«Web-программирование»** | |
|  | Выполнил:  Cтудент группы ТКИ-541,  Николаев А.М.  Проверил:  Доцент кафедры УиЗИ к.т.н.,  Сафронов А. И. |
| г. Москва - 2023 | |

ОГЛАВЛЕНИЕ

[Цель работы 3](#_Toc149515374)

[Формулировка задачи 3](#_Toc149515375)

[Технология локального подключения фреймворка *Vue.js* 4](#_Toc149515376)

[Таблица соответствия переменных и методов 4](#_Toc149515377)

[ХОД РАБОТЫ 5](#_Toc149515378)

[Содержательная часть по вставке циклической отрисовке 5](#_Toc149515379)

[Содержательная часть по наблюдателям 8](#_Toc149515380)

[ВЫВОД 13](#_Toc149515381)

Цель работы

Разработать локальное одностраничное веб-приложение (*LSPWA*) под управлением фреймворка *Vue.js* на языке *JavaScript* в соответствии с указаниями вариантов индивидуального задания (покрывают два раздела).

Формулировка задачи

Первая часть (Списки, перебор, *v-for*) для 12-го варианта:

Изобразить в области масштабируемой векторной графики <*svg*> размерности 1000 на 200 пикселей по ширине и высоте, соответственно, горизонтальную координатную ось со стрелкой, засечками, числовыми значениями под засечками и особой, более жирной засечкой в ноле (присутствует всегда, когда указан диапазон с различными знаками в начале и конце, даже в ситуациях, когда шаг засечек в точности не попадает в ноль), разграничивающей области отрицательных и положительных значений. Ось отстаёт от рамок <*svg*> на 100 пикселей, соответственно. Стрелка всегда фиксированных размеров, устанавливаемых автором (пользователь не управляет настройками вида этого элемента). Пользователь меняет шаг засечек, указывает начало и конец диапазона, указывает значение округления для текста маркеров под засечками посредством элементов <*input*>. В качестве основных элементов внутри <*svg*> использовать <*line*>, для маркеров, соответственно, использовать <*text*>.

Вторая часть (Наблюдатели *watch*) для 12-го варианта:

Заполняется статический маркированный список <*ul*>. Ввод новых значений предусмотрен через <*input*> по нажатии сочетания «*Ctrl*» + «*Enter*». Как только в поле <*input*> возникает значение-команда «#редактировать#» статический список меняется на функциональный <*select*>, в котором можно выбирать желаемую для редактирования позицию, её текст передаётся в <input>, где можно его отредактировать и по «*Ctrl*» + «*Enter*» изменить в перечне. Значение-команда «#вводить#» переводит список обратно в состояние статического, маркированного.

**Не использовать:** *jQuerry*, «чистый» *JavaScript,* каскадные таблицы стилей (*CSS*), директиву *v-for*, директивы *v-if*, *v-else*, *v-else-if*, *v-model*, теги <*template*>.

**Реализовать:**

– локально (*LSPWA*), не прибегая к инструментарию *Node.js* и *npm* (*Node Package Manager*);

– в файле с именем *index\_Familiya\_I\_O.html*

# Технология локального подключения фреймворка *Vue.js*

Vue.js можно подключить локально, добавив в ваш HTML следующий тег script:

<script src="https://cdn.jsdelivr.net/npm/vue@2"></script>

Этот тег подключает Vue.js напрямую из CDN, и нет необходимости устанавливать что-либо через Node.js или npm.

# Таблица соответствия переменных и методов

|  |  |  |
| --- | --- | --- |
| Переменная/Метод | Описание | Применение |
| start | Начало диапазона для координатной оси | Первое приложение |
| end | Конец диапазона для координатной оси | Первое приложение |
| step | Шаг засечек на координатной оси | Первое приложение |
| rounding | Значение округления для меток засечек | Первое приложение |
| ticksCount | Вычисляемое свойство для определения количества засечек | Первое приложение |
| tickPosition | Метод для определения позиции засечки | Первое приложение |
| tickLabel | Метод для определения метки засечки | Первое приложение |
| items | Список элементов | Второе приложение |
| inputValue | Значение поля ввода | Второе приложение |
| mode | Режим отображения (просмотр или редактирование) | Второе приложение |
| selectedItem | Выбранный элемент для редактирования | Второе приложение |
| handleInput | Метод для обработки ввода пользователя | Второе приложение |

# ХОД РАБОТЫ

## Содержательная часть по вставке циклической отрисовке

Код index\_Nikolaev\_A\_M\_1.html:

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>index\_Nikolaev\_A\_M\_1</title>

</head>

<body>

    <div id="app">

        <input v-model="start" placeholder="Начало диапазона">

        <input v-model="end" placeholder="Конец диапазона">

        <input v-model="step" placeholder="Шаг засечек">

        <input v-model="rounding" placeholder="Округление">

        <svg width="1000" height="200">

            <line x1="100" y1="100" x2="900" y2="100" stroke="black" />

            <line x1="900" y1="95" x2="910" y2="100" stroke="black" />

            <line x1="900" y1="105" x2="910" y2="100" stroke="black" />

            <line v-for="i in ticksCount"

            :x1="tickPosition(i)"

            y1="90"

            :x2="tickPosition(i)"

            y2="110"

            stroke="black" />

            <text v-for="i in ticksCount"

            :x="tickPosition(i)"

            y="80"

            font-size="10"

            text-anchor="middle">{{ tickLabel(i) }}</text>

        </svg>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/vue@2"></script>

    <script src="script1.js"></script>

</body>

</html>

Код script1.js:

new Vue({

    el: '#app',

    data: {

        start: -10,

        end: 10,

        step: 1,

        rounding: 2

    },

    computed: {

        ticksCount() {

            return Math.floor((this.end - this.start) / this.step) + 1;

        }

    },

    methods: {

        tickPosition(index) {

            const value = parseFloat(this.start) + index \* parseFloat(this.step);

            return 100 + (value - this.start) \* (800 / (this.end - this.start));

        },

        tickLabel(index) {

            const value = parseFloat(this.start) + index \* parseFloat(this.step);

            return parseFloat(value).toFixed(this.rounding);

        }

    }

});

|  |
| --- |
|  |
| Рисунок 1 – Вид страницы |

|  |
| --- |
|  |
| Рисунок 2 – Сеть Петри |

## Содержательная часть по наблюдателям

Код index\_Nikoilaev\_A\_M\_2.html:

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>index\_Nikolaev\_A\_M\_2</title>

</head>

<body>

    <div id="app2">

        <input v-model="inputValue" @keyup.ctrl.enter="handleInput">

        <ul v-if="mode === 'view'">

            <li v-for="item in items">{{ item }}</li>

        </ul>

        <select v-if="mode === 'edit'" v-model="selectedItem">

            <option v-for="item in items">{{ item }}</option>

        </select>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/vue@2"></script>

    <script src="script2.js"></script>

</body>

</html>

Код script2.js:

new Vue({

    el: '#app2',

    data: {

        items: ['Пример 1', 'Пример 2'],

        inputValue: '',

        mode: 'view',

        selectedItem: null

    },

    watch: {

        inputValue(value) {

            if (value === '#редактировать#') {

                this.mode = 'edit';

                this.inputValue = '';

            } else if (value === '#вводить#') {

                this.mode = 'view';

                this.inputValue = '';

            }

        }

    },

    methods: {

        handleInput() {

            if (this.mode === 'view') {

                this.items.push(this.inputValue);

            } else if (this.mode === 'edit') {

                const index = this.items.indexOf(this.selectedItem);

                if (index !== -1) {

                    this.items[index] = this.inputValue;

                }

            }

            this.inputValue = '';

        }

    }

});

|  |
| --- |
|  |
| Рисунок 3 – Начальная страница сайта |

|  |
| --- |
|  |
| Рисунок 4 – Ввод данных |

|  |
| --- |
|  |
| Рисунок 5 – Переключение режима |

|  |
| --- |
|  |
| Рисунок 6 – Выбор данных для изменения |

|  |
| --- |
|  |
| Рисунок 7 – Изменение данных |

|  |
| --- |
|  |
| Рисунок 5 – Сеть Петри |

ВЫВОД

В ходе выполнения работы была успешно достигнута поставленная цель - изучены и применены на практике основы работы с фреймворком Vue.js для создания локальных одностраничных веб-приложений (LSPWA).