|  |  |
| --- | --- |
| **Российский университет транспорта (МИИТ)**  **Институт транспортной техники и систем управления**  **Кафедра «Управление и защита информации»** | |
| **Отчёт**  **по практической работе**  **по теме «Технология локального подключения фреймворков»**  **по дисциплине «Web-программирование»**  **Вариант 16** | |
|  | Выполнили:  Студенты группы ТКИ-542  Шуриков Д. А.  Кожак И. А. |
|  | Проверил:  Доцент кафедры УиЗИ к.т.н. доцент.  Сафронов А. И. |
| Москва 2024 | |

1. Цель работы

Разработать локальные одностраничные *web*-приложение (*LSPWA*) под управлением фреймворка *Vue.js* на языке *JavaScript* в соответствии с указаниями вариантов индивидуального задания.

1. Формулировка задачи

**Реализовать:**

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

– *v-for* в файле с именем *index-v-for\_Familiya\_I\_O.html;*

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

– витки жизненного цикла в файле с именем *index-hooks\_Familiya\_I\_O.html.*

1. ПУНКТ 1 (WEB – страница 1)
   1. Формулировка задачи

Продумать схему тестирования и демонстрации работы всех витков / хуков / методов жизненного цикла фреймворка Vue.js в формате одностраничного web-приложения, отличающегося от рассмотренного в лекционном материале курса «Web-программирование». Реализовать схему тестирования строго под Vue.js 3.x (Приложение 1). Русифицировать схему жизненного цикла Vue.js 3.x (допускается использование готовой реализации из официальной документации).

* 1. Код web-приложения

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Hooks</title>

<script src="js/vue-3-5-13.js"></script>

</head>

<body>

<div id="app">

<h1>Hooks</h1>

<templates-selector v-if="!showEndMessage" :templates="templates" :selected-template="currentTemplate"></templates-selector>

<p v-else>Список завершен: {{ templates }}</p>

<button v-on:click="buttonClicked" v-if="!showEndMessage">{{ isLast ? 'Завершить' : 'Вперед' }}</button>

</div>

<script>

const {

createApp,

ref,

computed,

onBeforeMount,

onMounted,

onBeforeUpdate,

onUpdated,

onBeforeUnmount,

onUnmounted

} = Vue

const templates = ref({

'Ф.И.О.': '',

'Серия паспорта': '',

'Номер паспорта': '',

'Должность': '',

'Доля ставки': ''

})

const currentTemplate = ref(Object.keys(templates.value)[0])

const showEndMessage = ref(false)

const templatesSelector = {

template: `<p>{{ selectedTemplate }}: <input type="text" v-model="templates[selectedTemplate]"></p>`,

props: ['templates', 'selectedTemplate'],

setup() {

onBeforeMount(() => {

console.log('Child: onBeforeMount');

})

onMounted(() => {

console.log('Child: onMounted');

})

onBeforeUpdate(() => {

console.log('Child: onBeforeUpdate');

})

onUpdated(() => {

console.log('Child: onUpdated');

})

onBeforeUnmount(() => {

console.log('Child: onBeforeUnmount');

})

onUnmounted(() => {

console.log('Child: onUnmounted');

})

}

}

createApp({

components: {

templatesSelector

},

setup() {

const isLast = computed(() => {

const templatesNames = Object.keys(templates.value)

return templatesNames.indexOf(currentTemplate.value) === templatesNames.length - 1

})

const buttonClicked = () => {

const templatesNames = Object.keys(templates.value)

const currentIndex = templatesNames.indexOf(currentTemplate.value)

if (currentIndex < templatesNames.length - 1) currentTemplate.value = templatesNames[currentIndex + 1]

else showEndMessage.value = true

}

onBeforeMount(() => {

console.log('App: onBeforeMount');

})

onMounted(() => {

console.log('App: onMounted');

})

onBeforeUpdate(() => {

console.log('App: onBeforeUpdate');

})

onUpdated(() => {

console.log('App: onUpdated');

})

onBeforeUnmount(() => {

console.log('App: onBeforeUnmount');

})

onUnmounted(() => {

console.log('App: onUnmounted');

})

return {

templates,

currentTemplate,

buttonClicked,

isLast,

showEndMessage

}

}

}).mount("#app")

</script>

</body>

</html>

* 1. Результат отображения в браузере

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.1 Результат выполнения пункта 1

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.2 Результат выполнения пункта 1

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.3 Результат выполнения пункта 1

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.4 Результат выполнения пункта 1

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.5 Результат выполнения пункта 1

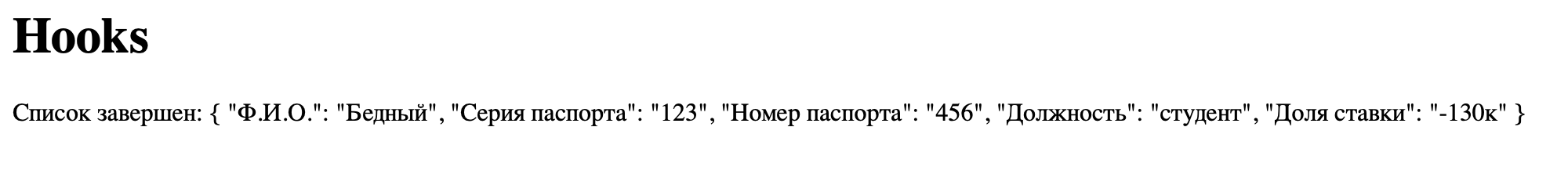


Рис.6 Результат выполнения пункта 1

* 1. Сеть Петри

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как круг, снимок экрана, черно-белый, луна

Автоматически созданное описание

Рис.7 Сеть Петри для пункта 1

1. ПУНКТ 2 (WEB – СТРАНИЦА 2)
   1. Формулировка задачи

Дан массив размерности [*4 x 7*], содержащий наименования изображений, планируемых к подстановке в ячейки таблицы. Все изображения подобрать одного размера, но различные по содержанию. Добиться эффекта, чтобы зазор между изображениями оказался минимален. Границы таблицы сделать невидимыми.

* 1. Код веб-приложения

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>v-for</title>

<script src="js/vue-3-5-13.js"></script>

</head>

<body>

<div id="app">

<h1>v-for</h1>

<table cellpadding="0" cellspacing="0">

<tr v-for="imagesRow in imagesData" style="line-height: 0px">

<td v-for="image in imagesRow">

<img :src="image" alt=""/>

</td>

</tr>

</table>

</div>

<script>

const {

createApp,

ref

} = Vue

const imagesData = ref([

[ 'images/image\_part\_001.png', 'images/image\_part\_002.png', 'images/image\_part\_003.png', 'images/image\_part\_004.png', 'images/image\_part\_005.png', 'images/image\_part\_006.png', 'images/image\_part\_007.png' ],

[ 'images/image\_part\_008.png', 'images/image\_part\_009.png', 'images/image\_part\_010.png', 'images/image\_part\_011.png', 'images/image\_part\_012.png', 'images/image\_part\_013.png', 'images/image\_part\_014.png' ],

[ 'images/image\_part\_015.png', 'images/image\_part\_016.png', 'images/image\_part\_017.png', 'images/image\_part\_018.png', 'images/image\_part\_019.png', 'images/image\_part\_020.png', 'images/image\_part\_021.png' ],

[ 'images/image\_part\_022.png', 'images/image\_part\_023.png', 'images/image\_part\_024.png', 'images/image\_part\_025.png', 'images/image\_part\_026.png', 'images/image\_part\_027.png', 'images/image\_part\_028.png' ]

])

createApp({

setup() {

return {

imagesData

}

}

}).mount("#app")

</script>

</body>

</html>

* 1. Результат отображения в браузере

Изображение выглядит как защитные очки, солнцезащитные очки, улыбка, очки

Автоматически созданное описание

Рис.8 Результат выполнения пункта 2

1. Пункт 3 (web-страница 3)
   1. Формулировка задачи

Создать конвертер температур. Любая вводимая величина влияет на остальные. Отдельных кнопок конвертации не предусмотрено. В процессе конвертации участвуют: цельсии, кельвины, рёмеры, реомюры, ньютоны, делили и фаренгейты. Оформить конвертер таблично.

* 1. Код веб-приложения

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Watches</title>

<script src="js/vue-3-5-13.js"></script>

</head>

<body>

<div id="app">

<h1>Watches</h1>

<p> Цельсии: <input v-model="c"/></p>

<p> Кельвины: <input v-model="k"/></p>

<p> Рёмеры: <input v-model="rem"/></p>

<p> Реомюры: <input v-model="reo"/></p>

<p> Ньютоны: <input v-model="n"/></p>

<p> Делили: <input v-model="d"/></p>

<p> Фаренгейты: <input v-model="f"/></p>

</div>

<script>

const {

createApp,

ref,

watch

} = Vue

const round = (val) => Math.round(val\*1000)/1000

const checks = (newVal, oldVal) => newVal !== oldVal && Number(newVal)

const fromCelsiusFormulas = {

'c': (val) => round(Number(val)),

'k': (val) => round(Number(val) + 273.15),

'rem': (val) => round(Number(val)\*0.525 + 7.5),

'reo': (val) => round(Number(val) \* 0.8),

'n': (val) => round(Number(val) \* 0.33),

'd': (val) => round((100 - Number(val)) \* 1.5),

'f': (val) => round((Number(val) \* 1.8) + 32)

}

const toCelsiusFormulas = {

'c': (val) => round(Number(val)),

'k': (val) => round(Number(val) - 273.15),

'rem': (val) => round((Number(val) - 7.5)/0.525),

'reo': (val) => round(Number(val) / 0.8),

'n': (val) => round(Number(val) / 0.33),

'd': (val) => round(Number(val) / 1.5 + 100),

'f': (val) => round((Number(val) - 32) / 1.8)

}

const c = ref(0)

const k = ref(0)

const rem = ref(0)

const reo = ref(0)

const n = ref(0)

const d = ref(0)

const f = ref(0)

watch(c, (newValue, oldValue) => {

if (!checks(newValue, oldValue)) return

k.value = fromCelsiusFormulas['k'](c.value)

rem.value = fromCelsiusFormulas['rem'](c.value)

reo.value = fromCelsiusFormulas['reo'](c.value)

n.value = fromCelsiusFormulas['n'](c.value)

d.value = fromCelsiusFormulas['d'](c.value)

f.value = fromCelsiusFormulas['f'](c.value)

})

watch([k], (newValue, oldValue) => {

if (!checks(newValue, oldValue)) return

c.value = toCelsiusFormulas['k'](newValue)

})

watch([rem], (newValue, oldValue) => {

if (!checks(newValue, oldValue)) return

c.value = toCelsiusFormulas['rem'](newValue)

})

watch([reo], (newValue, oldValue) => {

if (!checks(newValue, oldValue)) return

c.value = toCelsiusFormulas['reo'](newValue)

})

watch([n], (newValue, oldValue) => {

if (!checks(newValue, oldValue)) return

c.value = toCelsiusFormulas['n'](newValue)

})

watch([d], (newValue, oldValue) => {

if (!checks(newValue, oldValue)) return

c.value = toCelsiusFormulas['d'](newValue)

})

watch([f], (newValue, oldValue) => {

if (!checks(newValue, oldValue)) return

c.value = toCelsiusFormulas['f'](newValue)

})

c.value = 25

createApp({

setup() {

return {

c,

k,

rem,

reo,

n,

d,

f

}

}

}).mount("#app")

</script>

</body>

</html>

* 1. Результат отображения в браузере

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис.9 Результат выполнения пункта 3

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис.10 Результат выполнения пункта 3

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис.11 Результат выполнения пункта 3

* 1. Сеть Петри

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, круг, Шрифт

Автоматически созданное описание

Рис.12 Сеть Петри для пункта 3

1. Спецификация оборудования, на котором выполнялась работа.

Таблица 1 –Cпецификация параметров операционной системы

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Характеристика** | **Сведения** |
| **1** | Тип устройства | Ноутбук |
| **2** | Модель | MacBook Air 2020 |
| **3** | Операционная система | *Ventura 13.0* |
| **4** | Процессор | *Intel Core i3* |
| **5** | Оперативная память | 8 Гб |
| **6** | Объём жёсткого диска | 256 Гб |

Изображение выглядит как текст, снимок экрана, логотип, мультимедиа

Автоматически созданное описание

Рисунок 13 – Cведения о браузере *Safari*

1. Схема жизненного цикла (на русском языке) одностраничного *web*-приложения под управлением *Vue.js 2.x.*

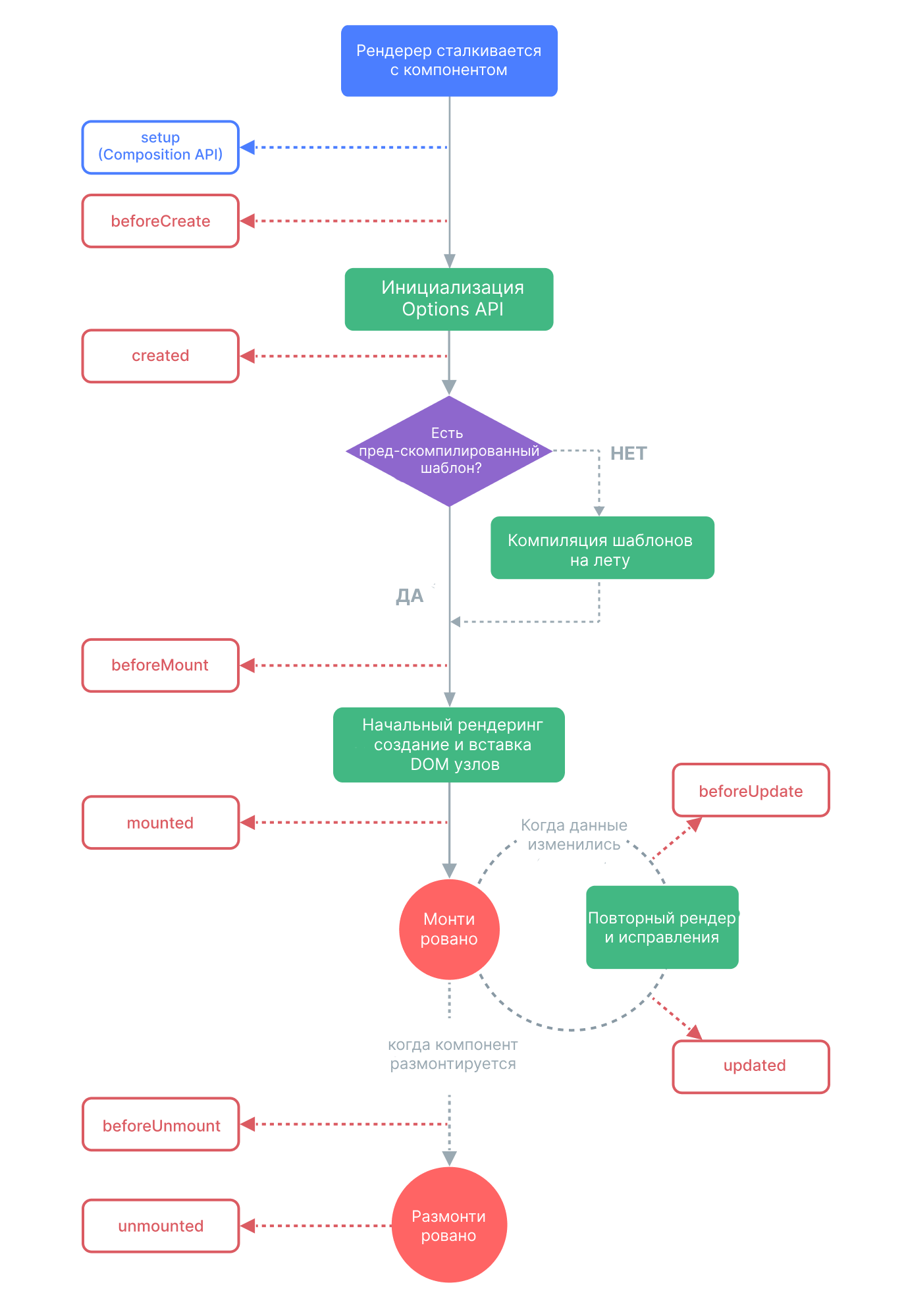


Рисунок 14 – Витки жизненного цикла экземпляра *Vue 2.x*

1. Таблица соотвествия переменных и методов, используемых в web-приложении

|  |  |  |
| --- | --- | --- |
| Задание | Переменная/метод | Описание |
| 1 | logLifecycle() | Регистрация событий жизненного цикла |
| 1 | beforeCreate() | Хук жизненного цикла, вызывается до инициализации данных компонента |
| 1 | created() | Хук жизненного цикла, вызывается после инициализации данных компонента |
| 1 | beforeMount() | Хук жизненного цикла, вызывается перед монтированием компонента в DOM |
| 1 | mounted() | Хук жизненного цикла, вызывается после того, как компонент смонтирован в DOM |
| 1 | beforeUpdate() | Хук жизненного цикла, вызывается перед обновлением компонента |
| 1 | updated() | Хук жизненного цикла, вызывается после обновления компонента в DOM |
| 1 | beforeUnmount() | Хук жизненного цикла, вызывается перед размонтированием компонента |
| 1 | unmounted() | Хук жизненного цикла, вызывается после удаления компонента из DOM |

1. Вывод

Мы получили навыки по созданию локальных одностраничных web-приложение под управлением фреймворка Vue.js на языке JavaScript.