|  |  |
| --- | --- |
| **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 13** | |
|  | Выполнили:  Студенты группы ТКИ-542  Зинченко Б.А.  Круглов В.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="https://unpkg.com/vue@3/dist/vue.global.js"></script>

</head>

<body>

<div id="app">

<h1>{{ message }}</h1>

<button @click="updateMessage">Update Message</button>

<button @click="destroyComponent">Destroy Component</button>

<div v-if="showComponent">

<child-component></child-component>

</div>

</div>

<script>

const {

createApp,

ref,

onBeforeMount,

onMounted,

onBeforeUpdate,

onUpdated,

onBeforeUnmount,

onUnmounted

} = Vue

const ChildComponent = {

template: `<p>Child Component</p>`,

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: {

ChildComponent

},

setup() {

const message = ref('message');

const showComponent = ref(true);

const updateMessage = () => {

message.value = 'Message updated!';

};

const destroyComponent = () => {

showComponent.value = false;

};

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 {

message,

showComponent,

updateMessage,

destroyComponent

}

}

}).mount("#app")

</script>

</body>

</html>

* 1. Результат отображения в браузере

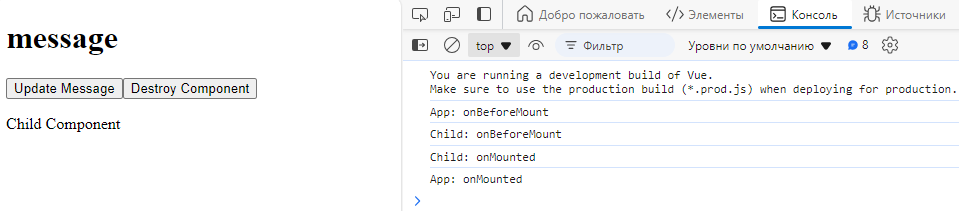


Рис.1 Результат выполнения пункта 1

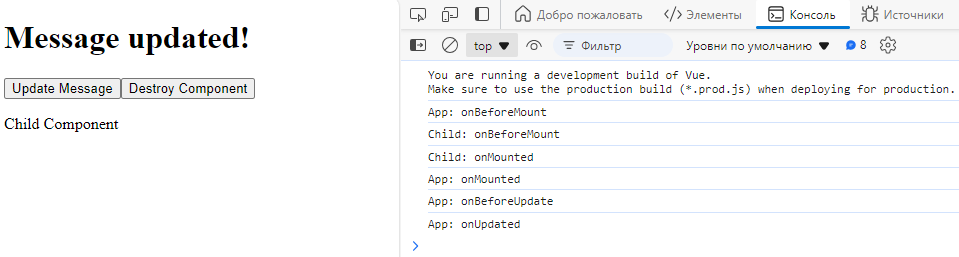


Рис.2 Результат выполнения пункта 1

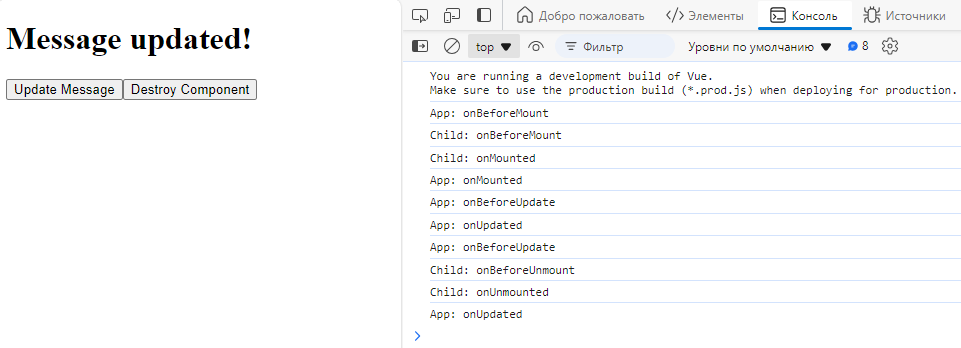


Рис.3 Результат выполнения пункта 1

* 1. Сеть Петри



Рис.4 Сеть Петри для пункта 1

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

Изобразить внутри обрамлённой области масштабируемой векторной графики <*svg*> с фиксированными значениями ширины и высоты функцию вида *y(x) = sin(x)* окружностями <*circle*> настраиваемого радиуса (2 – 10 пикселей с единичным шагом в <*input*> в режиме *range*) на белом фоне без осей. Предоставить оператору возможность масштабирования с единичным шагом функции в фиксированном диапазоне 10%–100% через <*input*> в режиме *range*.

* 1. Код веб-приложения

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <title>График функции с использованием v-for</title>

    <script src="https://cdn.jsdelivr.net/npm/vue@2.6.14"></script>

    <style>

        svg {

            border: 1px solid #000;

            background-color: #fff;

        }

    </style>

</head>

<body>

<div id="app">

    <label>Радиус круга (2-10):</label>

    <input type="range" v-model="radius" min="2" max="10">

    <label>Масштаб (10%-100%):</label>

    <input type="range" v-model="scale" min="10" max="100">

    <svg :width="width" :height="height">

        <circle v-for="point in points" :cx="point.x \* scaleFactor" :cy="point.y \* scaleFactor"

                :r="radius" fill="blue"></circle>

    </svg>

</div>

<script>

    new Vue({

        el: '#app',

        data() {

            return {

                radius: 5,

                scale: 100,

                width: 400,

                height: 200,

                points: Array.from({ length: 50 }, (\_, i) => ({

                    x: i \* 8,

                    y: Math.sin(i \* 0.2) \* 50 + 100

                }))

            };

        },

        computed: {

            scaleFactor() {

                return this.scale / 100;

            }

        }

    });

</script>

</body>

</html>

* 1. Результат отображения в браузере

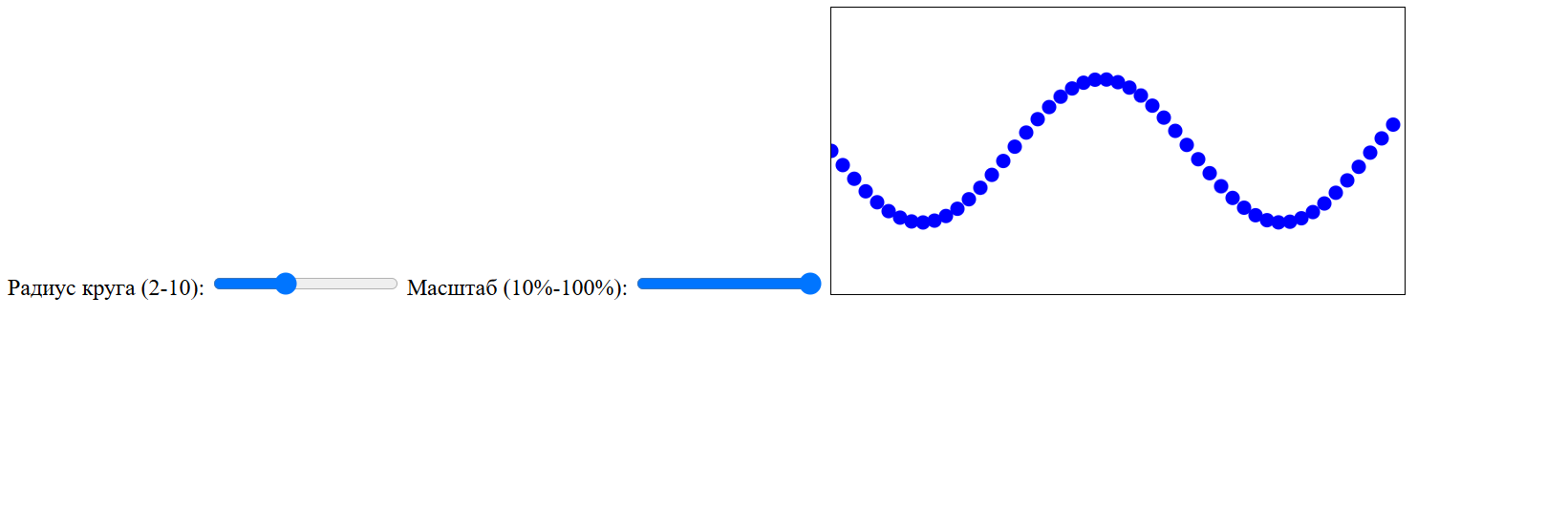


Рис.5 Результат выполнения пункта 2

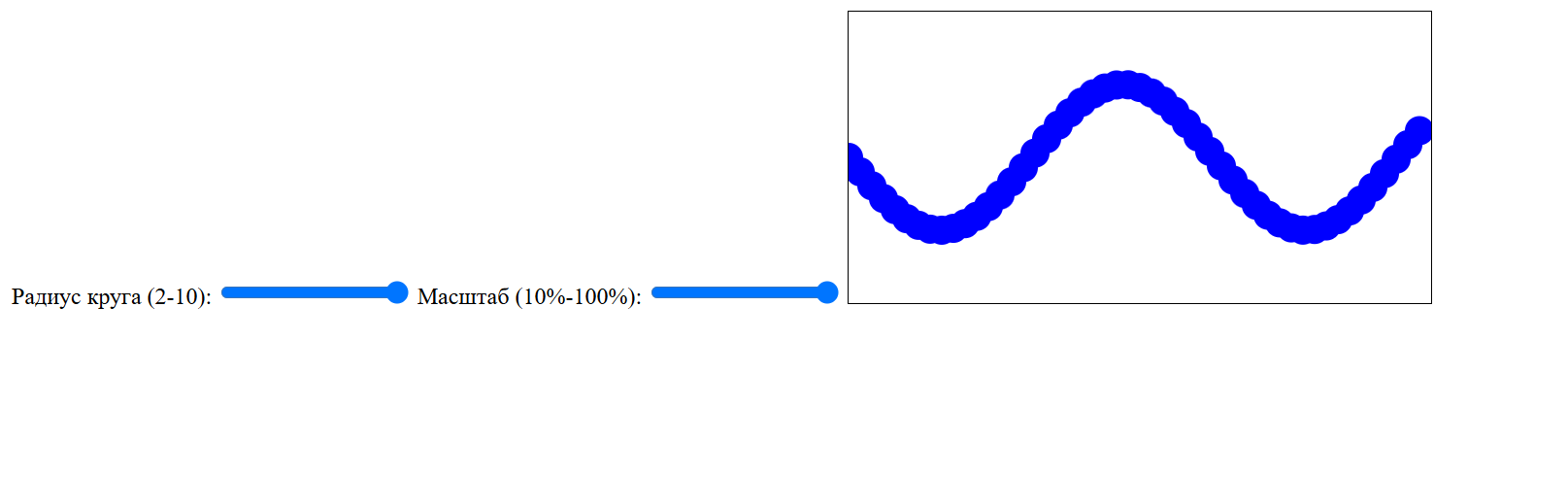


Рис.6 Результат выполнения пункта 2

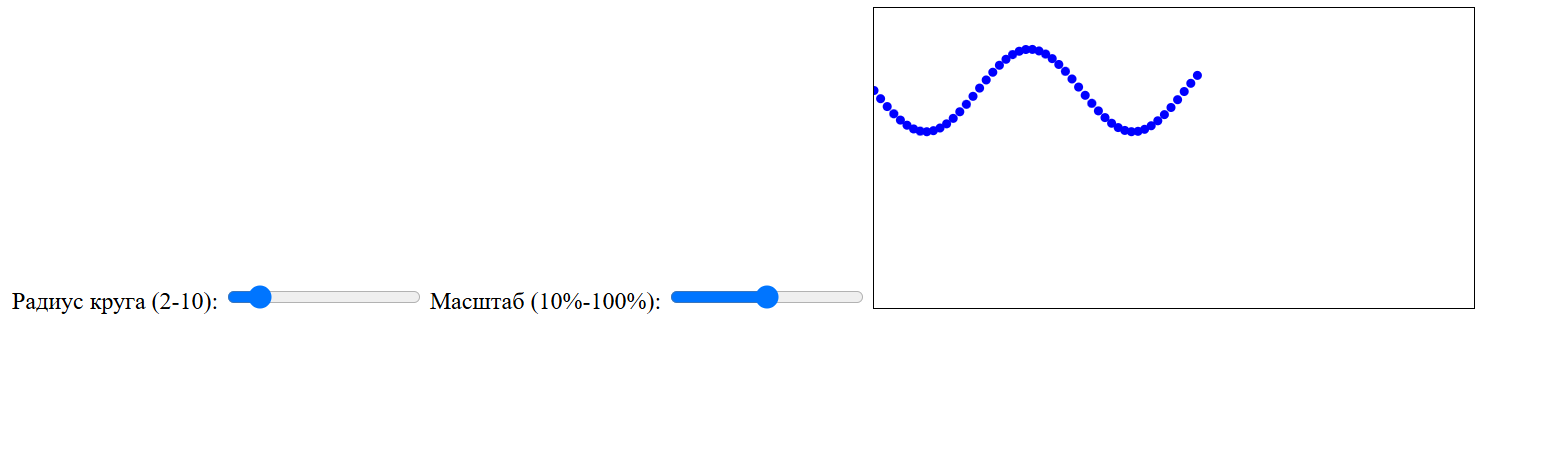


Рис.7 Результат выполнения пункта 2

* 1. Сеть Петри



Рис.8 Сеть Петри для пункта 2

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

Создать конвертер валют. Любая вводимая величина влияет на остальные. Отдельных кнопок конвертации не предусмотрено. В процессе конвертации участвуют рубли, доллары, евро, йены, юани, рупии. Взять к рассмотрению курс, зафиксированный на момент: 10 января 2024 года. Оформить конвертер таблично.

* 1. Код веб-приложения

<!DOCTYPE html>

<html lang="ru">

<head>

  <meta charset="UTF-8">

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

  <title>Конвертер валют на Vue.js</title>

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

  <style>

    table { width: 100%; border-collapse: collapse; }

    th, td { border: 1px solid #ccc; padding: 10px; text-align: center; }

    input { width: 100%; }

  </style>

</head>

<body>

  <div id="app">

    <h2>Конвертер валют (Курс от 10 января 2024 года)</h2>

    <table>

      <tr>

        <th>Валюта</th>

        <th>Сумма</th>

      </tr>

      <tr v-for="(value, currency) in currencies" :key="currency">

        <td>{{ currencyNames[currency] }}</td>

        <td><input type="number" v-model.number="currencies[currency]" placeholder="Введите сумму"></td>

      </tr>

    </table>

  </div>

  <script>

    new Vue({

      el: '#app',

      data: {

        // Курсы валют относительно рубля на 10 января 2024 года

        exchangeRates: {

          rub: 1,

          usd: 75.50,

          eur: 82.30,

          jpy: 0.55,

          cny: 10.50,

          inr: 0.92

        },

        currencyNames: {

          rub: 'Рубль (RUB)',

          usd: 'Доллар США (USD)',

          eur: 'Евро (EUR)',

          jpy: 'Йена (JPY)',

          cny: 'Юань (CNY)',

          inr: 'Рупия (INR)'

        },

        // Значения валют для пересчета

        currencies: {

          rub: null,

          usd: null,

          eur: null,

          jpy: null,

          cny: null,

          inr: null

        },

        updatingCurrency: null

      },

      watch: {

        'currencies.rub'(newValue) { this.updateCurrency('rub', newValue); },

        'currencies.usd'(newValue) { this.updateCurrency('usd', newValue); },

        'currencies.eur'(newValue) { this.updateCurrency('eur', newValue); },

        'currencies.jpy'(newValue) { this.updateCurrency('jpy', newValue); },

        'currencies.cny'(newValue) { this.updateCurrency('cny', newValue); },

        'currencies.inr'(newValue) { this.updateCurrency('inr', newValue); }

      },

      methods: {

        updateCurrency(inputCurrency, value) {

      if (this.updatingCurrency === inputCurrency || isNaN(value) || value === null || value === '') return;

        this.updatingCurrency = inputCurrency;

    for (const currency in this.currencies) {

      if (currency === inputCurrency) {

      this.currencies[currency] = value;

    } else {

      const convertedValue = (parseFloat(value) / this.exchangeRates[inputCurrency]) \* this.exchangeRates[currency];

    this.currencies[currency] = parseFloat(convertedValue.toFixed(2));

      }

    }

    this.updatingCurrency = null;

    }

  }

    });

  </script>

</body>

</html>

* 1. Результат отображения в браузере



Рис.9 Результат выполнения пункта 3

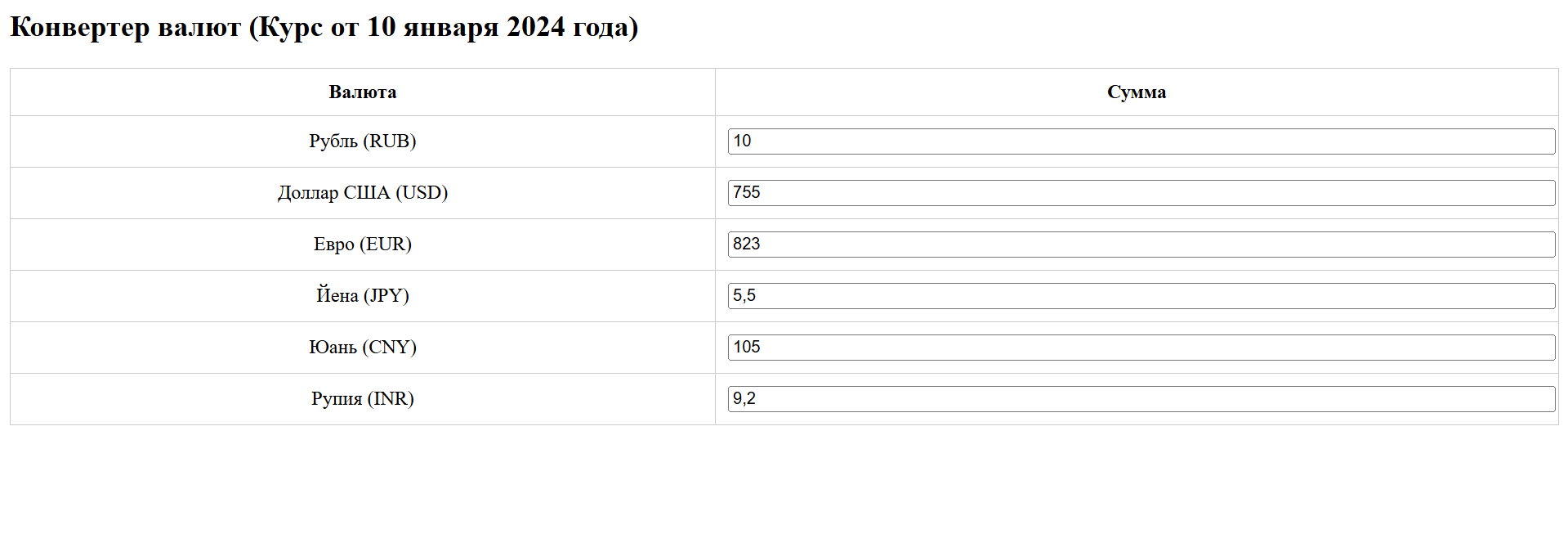


Рис.10 Результат выполнения пункта 3

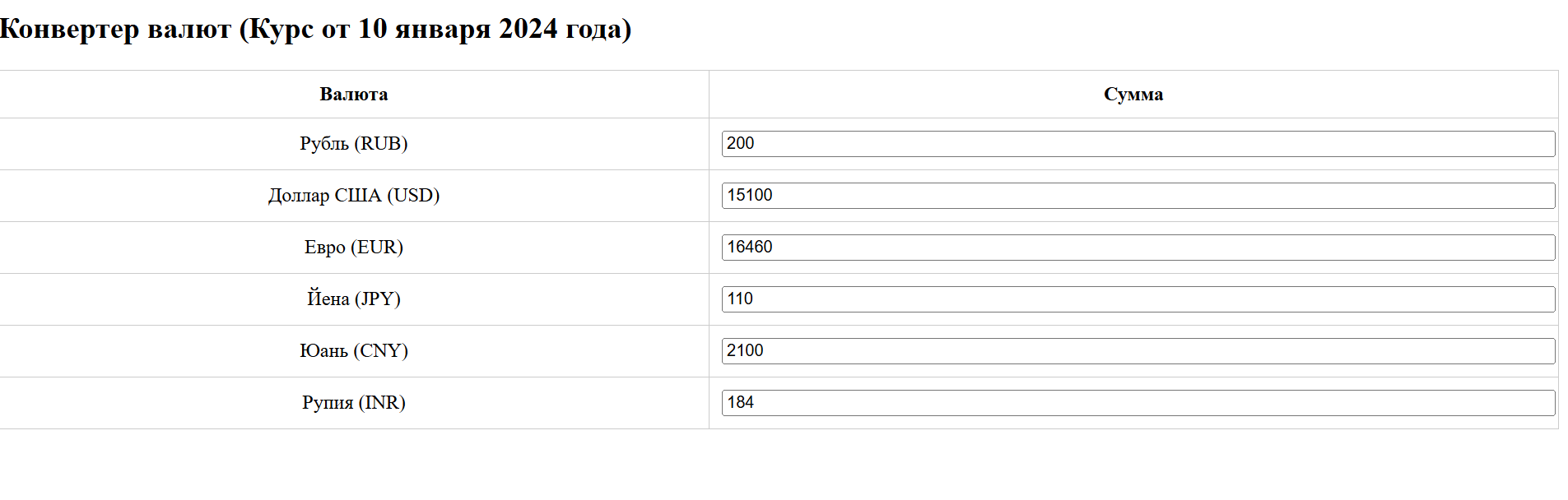


Рис.11 Результат выполнения пункта 3

* 1. Сеть Петри



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 | Lenovo thinkpad t14 |
| **3** | Операционная система | *Microsoft Windows 11* |
| **4** | Процессор | *AMD Ryzen 5* |
| **5** | Оперативная память | 32 Гб |
| **6** | Объём жёсткого диска | 1 Тб |

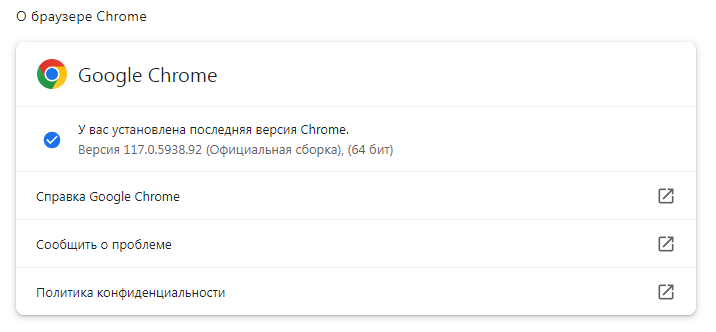


Рисунок 13 – Cведения о браузере *Google Chrome*

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

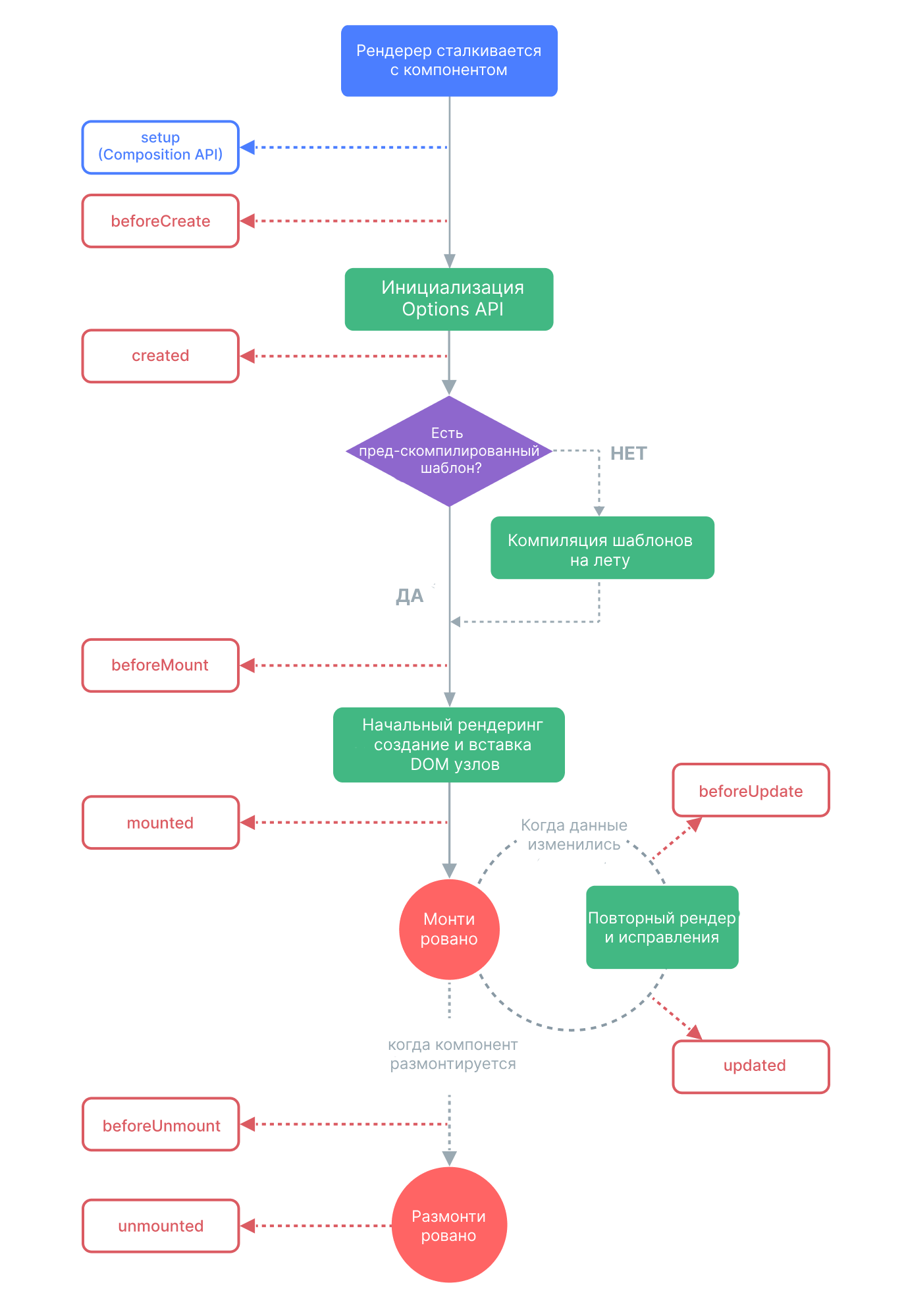


Рисунок 14 – Витки жизненного цикла экземпляра *Vue 2.x*

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.