

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ (МИИТ)**

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Кафедра «Управление и защита информации»

**Отчет по лабораторной работе**

«**Линейный и разветвляющийся вычислительные процессы**»

**по дисциплине**

«Web программирование»

Вариант 10

**Выполнил:** студенты

группы ТКИ-541

Плуталов Е.А., Мацко А.А.

**Проверил:** доцент кафедры УиЗИ, к.т.н. Сафронов А.И.

**Москва 2024 г.**

Оглавление

[Цель работы 3](#_Toc180318033)

[Постановка задачи 3](#_Toc180318034)

[Технология локального подключения фреймворка Vue.js 4](#_Toc180318035)

[Таблица соответствия переменных и методов, используемых в web-приложениях 5](#_Toc180318036)

[Содержательная часть по параметрической вставке 7](#_Toc180318037)

[Содержательная часть по параметрической связке 13](#_Toc180318038)

[Содержательная часть по методам и событиям 17](#_Toc180318039)

[Содержательная часть по условной отрисовке 23](#_Toc180318040)

[Вывод 35](#_Toc180318041)

# Цель работы

Цель работы заключается в создании локальных web-приложений на Vue.js с использованием параметрических вставок, связок, методов и событий, а также условной отрисовки. Разработка будет выполнена без использования Node.js и npm, с акцентом на локальное подключение фреймворка Vue.js.

# Постановка задачи

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

* **Задание 1:**

Настроить при помощи параметрической подстановки тег экранного (выпадающего) списка <select></select>. Пункты списка записать в массив. Размер массива параметрически задаёт количество отображаемых элементов <select></select>. В <option></option> в «усатых скобках» подставлять элементы массива. Для наглядной демонстрации влияния подставляемых значений параметров предусмотреть 7 элементов массива, из которых в <option></option> разместить только 5.

* **Задание 2:**

. Предусмотреть элемент ввода данных <input></input> в режиме ввода текста («text»). При наличии фокуса в интерфейсном элементе управления отработать сочетания клавиш следующим образом:

– «Ctrl» + «P» – не вызывать диалоговое окно печати страницы, а вызывать вместо него диалоговое окно «Prompt» с вопросом «Вывести диалоговое окно печати?». Принимаются строковые ответы вида: только «да» или «нет» (с учётом регистра). Все иные значения, введённые в предусмотренное стандартом поле, а также отмена вызванного диалога «Prompt» приводят к повторному вызову диалога «Prompt» до тех пор, пока не будет получен необходимый ответ: либо «да», либо «нет», записанных строчными буквами (с учётом регистра).

* **Задание 3:**

На web-страницу выводится активный (для взаимодействия с пользователем) экранный список из семи элементов <select></select> в режиме множественного выбора (multiple). Под списком расположены 7 элементов <input></input> в текстовом режиме. При выделении позиций в списке они последовательно заполняют свободные элементы <input></input>. Предусмотреть сброс выбранных значений по утере фокуса списком <select></select>.

* **Задание 4:**

Продумать опросник на произвольно выбранную тематику, состоящий из десяти вопросов. Веб-приложение содержит стартовый экран, приглашающий к началу тестирования по кнопке «Начать тестирование». Для каждого вопроса предлагаются четыре варианта ответа, среди которых всегда один и только один является верным. В качестве активных элементов управления использовать опции (radiobuttons). Предусмотреть систему из двух подсказок, активируемых и демонстрируемых по необходимости (по галочке checkbox): а) пояснение с наводкой на правильный ответ (сокращает результат на 0,15 балла); б) устранение из перечня двух заведомо ложных вариантов (сокращает результат на 0,5 баллов). В самом начале демонстрируется всегда первый вопрос. Пользователь может в любой момент перейти к любому вопросу по соответствующей кнопке, имитирующей переход к странице по вкладке (не по ссылке). По итогам нажатия на кнопку «Выполнено» выдавать результат успеха прохождения опроса в процентах. Подсчёт результатов: 1 верный ответ без взятия подсказок – 1 балл. При взятии обеих подсказок «штрафы» суммируются. Экран вывода результатов отдельный по завершении тестирования без возможности возврата к тестированию.

# Технология локального подключения фреймворка Vue.js

Vue.js подключается локально через ссылку на загруженный файл `vue.global.js`. Этот файл добавляется в `<script>` в HTML-документе**:**

<script src="vue.global.js"></script>

# Содержательная часть по параметрической вставке

|  |
| --- |
| <html>  <head></head>  <body>  <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>  <div id="app">  <select :size="size\_arr">  <option>{{ items[0] }}</option>  <option>{{ items[1] }}</option>  <option>{{ items[2] }}</option>  <option>{{ items[3] }}</option>  <option>{{ items[4] }}</option>  </select>  </div>  <script>  const { createApp, ref } = Vue  createApp({  setup() {  const size\_arr = ref('5')  const items = ref(['1', '2', '3', '4', '5', '6', '7'])    return {  items,  size\_arr  }  }  }).mount('#app')  </script>  </body>  </html> |
| Код программы Задание 1 |

|  |
| --- |
|  |
| Рисунок 1 – Результат работы программы |

# Содержательная часть по параметрической связке

|  |
| --- |
| <html>  <head></head>  <body>  <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>  <div id="app" @click.right.prevent>  <input @click.left.ctrl="prnt"  @keydown.ctrl.p.prevent="handlePrintDialog"  @keydown.ctrl.з.prevent="handlePrintDialog">  </div>  <script>  const { createApp } = Vue;  createApp({  data() {  return {  key: "",  };  },  methods: {  prnt() {  alert("Настройка печати");  },  handlePrintDialog(e) {  const askPrint = () => {  let response = prompt("Вывести диалоговое окно печати? (да/нет)");    if (response === "да") {  window.print();  } else if (response === "нет") {  alert("Печать отменена");  } else {  askPrint(); // Повторный вызов диалога при неверном ответе  }  };  askPrint();  }  }  }).mount('#app');  </script>  </body>  </html> |
| Код программы Задание 2 |

|  |
| --- |
|  |
| Рисунок 2 – Результат работы программы |
|  |
|  |

# Содержательная часть по методам и событиям

|  |
| --- |
| <html>  <head>  <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>  </head>  <body>  <div id="app">  <!-- Элемент select с режимом множественного выбора -->  <select multiple size="7" @blur="resetSelections" @change="handleSelectionChange">  <option value="Option 1">Option 1</option>  <option value="Option 2">Option 2</option>  <option value="Option 3">Option 3</option>  <option value="Option 4">Option 4</option>  <option value="Option 5">Option 5</option>  <option value="Option 6">Option 6</option>  <option value="Option 7">Option 7</option>  </select>  <!-- Семь текстовых полей -->  <input type="text" v-bind:value="inputs[0]" readonly />  <input type="text" v-bind:value="inputs[1]" readonly />  <input type="text" v-bind:value="inputs[2]" readonly />  <input type="text" v-bind:value="inputs[3]" readonly />  <input type="text" v-bind:value="inputs[4]" readonly />  <input type="text" v-bind:value="inputs[5]" readonly />  <input type="text" v-bind:value="inputs[6]" readonly />  </div>  <script>  const { createApp } = Vue;  createApp({  data() {  return {  inputs: ['', '', '', '', '', '', ''], // Массив для значений текстовых полей (7 элементов)  };  },  methods: {  handleSelectionChange(event) {  // Получение выбранных значений  const selectedOptions = Array.from(event.target.selectedOptions).map(option => option.value);  // Очистка текстовых полей  for (let i = 0; i < 7; i++) {  this.inputs[i] = selectedOptions[i] || ''; // Заполняем поля или оставляем пустыми  }  },  // Сброс выбранных значений при потере фокуса  resetSelections(event) {  event.target.selectedIndex = -1; // Сбрасываем выбранные опции  this.inputs = ['', '', '', '', '', '', '']; // Очищаем текстовые поля  }  }  }).mount('#app');  </script>  </body>  </html> |
| Код программы Задание 3 |

|  |
| --- |
|  |
| Рисунок 3 – Результат работы программы |
|  |
| Рисунок 5 – Результат работы программы |
|  |
| Рисунок 6 – Сеть Петри Задание 3 |

# Содержательная часть по условной отрисовке

|  |
| --- |
| <html>  <head>  <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>  </head>  <body>  <div id="app">  <div v-if="!testStarted && !testFinished">  <!-- Стартовый экран -->  <h1>Добро пожаловать в опрос!</h1>  <button @click="startTest">Начать тестирование</button>  </div>  <div v-if="testStarted && !testFinished">  <!-- Вопросы -->  <div>  <h2>{{ questions[currentQuestion].question }}</h2>  <!-- Опции с radiobuttons -->  <label><input type="radio" v-model="selectedAnswers[currentQuestion]" value="0"> {{ questions[currentQuestion].answers[0] }}</label><br>  <label><input type="radio" v-model="selectedAnswers[currentQuestion]" value="1"> {{ questions[currentQuestion].answers[1] }}</label><br>  <label><input type="radio" v-model="selectedAnswers[currentQuestion]" value="2"> {{ questions[currentQuestion].answers[2] }}</label><br>  <label><input type="radio" v-model="selectedAnswers[currentQuestion]" value="3"> {{ questions[currentQuestion].answers[3] }}</label><br>  <!-- Подсказки -->  <div>  <label><input type="checkbox" v-model="hintsUsed[currentQuestion].hint1"> Пояснение с наводкой (минус 0.15 балла)</label><br>  <label><input type="checkbox" v-model="hintsUsed[currentQuestion].hint2"> Устранить два ложных варианта (минус 0.5 балла)</label><br>  </div>  <!-- Кнопки перехода между вопросами -->  <div>  <button v-if="currentQuestion > 0" @click="previousQuestion">Назад</button>  <button v-if="currentQuestion < 9" @click="nextQuestion">Далее</button>  <button v-if="currentQuestion === 9" @click="finishTest">Выполнено</button>  </div>  </div>  <!-- Вкладки для переключения вопросов -->  <div>  <button v-for="i in 10" :key="i" @click="goToQuestion(i - 1)">Вопрос {{ i }}</button>  </div>  </div>  <!-- Экран с результатом -->  <div v-if="testFinished">  <h1>Тест завершён!</h1>  <p>Ваш результат: {{ finalScore }}%</p>  </div>  </div>  <script>  const { createApp } = Vue;  createApp({  data() {  return {  testStarted: false, // Флаг, начато ли тестирование  testFinished: false, // Флаг, завершено ли тестирование  currentQuestion: 0, // Текущий вопрос  selectedAnswers: Array(10).fill(null), // Выбранные ответы на вопросы  hintsUsed: Array(10).fill({ hint1: false, hint2: false }), // Использованные подсказки  questions: [  {  question: "Вопрос 1: Что такое Vue.js?",  answers: ["Библиотека", "Фреймворк", "Язык", "Редактор"],  correctAnswer: 1  },  {  question: "Вопрос 2: Сколько времени в часе?",  answers: ["50 минут", "60 минут", "40 минут", "30 минут"],  correctAnswer: 1  },  {  question: "Вопрос 3: Какой язык используется для веб-разработки?",  answers: ["HTML", "Python", "Java", "C++"],  correctAnswer: 0  },  {  question: "Вопрос 4: Какая планета самая большая в Солнечной системе?",  answers: ["Земля", "Марс", "Юпитер", "Венера"],  correctAnswer: 2  },  {  question: "Вопрос 5: Кто написал 'Войну и мир'?",  answers: ["Достоевский", "Толстой", "Пушкин", "Чехов"],  correctAnswer: 1  },  {  question: "Вопрос 6: Сколько континентов на Земле?",  answers: ["5", "6", "7", "8"],  correctAnswer: 2  },  {  question: "Вопрос 7: Какая химическая формула воды?",  answers: ["H2O", "CO2", "O2", "NaCl"],  correctAnswer: 0  },  {  question: "Вопрос 8: В каком году началась Вторая мировая война?",  answers: ["1914", "1939", "1941", "1945"],  correctAnswer: 1  },  {  question: "Вопрос 9: Какой цвет получается при смешивании синего и жёлтого?",  answers: ["Зелёный", "Оранжевый", "Фиолетовый", "Коричневый"],  correctAnswer: 0  },  {  question: "Вопрос 10: Столица Франции?",  answers: ["Мадрид", "Лондон", "Париж", "Берлин"],  correctAnswer: 2  }  ],  finalScore: 0 // Итоговый результат  };  },  methods: {  startTest() {  this.testStarted = true;  },  nextQuestion() {  if (this.currentQuestion < 9) {  this.currentQuestion++;  }  },  previousQuestion() {  if (this.currentQuestion > 0) {  this.currentQuestion--;  }  },  goToQuestion(index) {  this.currentQuestion = index;  },  finishTest() {  let totalScore = 0;  for (let i = 0; i < 10; i++) {  let baseScore = 1; // Один балл за правильный ответ  // Если использована первая подсказка, вычитаем 0.15 балла  if (this.hintsUsed[i].hint1) {  baseScore -= 0.15;  }  // Если использована вторая подсказка, вычитаем 0.5 балла  if (this.hintsUsed[i].hint2) {  baseScore -= 0.5;  }  // Проверка, был ли правильный ответ  if (parseInt(this.selectedAnswers[i]) === this.questions[i].correctAnswer) {  totalScore += baseScore;  }  }  // Вычисляем процент успеха  this.finalScore = Math.round((totalScore / 10) \* 100);  // Завершаем тестирование  this.testFinished = true;  this.testStarted = false;  }  }  }).mount('#app');  </script>  </body>  </html> |
| Код программы Задание 4 |

|  |
| --- |
|  |
| Рисунок 7 – Результат работы программы |
|  |
| Рисунок 8 – Результат работы программы |
| Рисунок 9 – Результат работы программы |
| Рисунок 10 – Результат работы программы |
|  |
| Рисунок 11 – Сеть Петри Задание 4 |

# Вывод

В данной работе были получены навыки по созданию локальных web-приложений на Vue.js, изучены основные понятия, принципы и методы параметрических вставок, связок, методов и событий, а также условной отрисовки. Было выполнено создание локального приложения без использования Node.js и npm, с акцентом на локальное подключение фреймворка Vue.js. Поставленные задачи выполнены в полном объёме.