|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Российский университет транспорта (МИИТ)**  **Институт транспортной техники и систем управления**  **Кафедра «Управление и защита информации»** | | | **Задание №3**  **по теме «Технология локального подключения фреймворков»**  **по дисциплине «Web-программирование»** | | |  | Выполнил:  Студент группы ТКИ-542  Волочинский И.О.  Ефремов Д.Ю.  Проверил:  Доцент кафедры УиЗИ, к.т.н., с.н.с  Сафронов А.И. | | Москва 2024 | | |

Оглавление

[1. Цель работы 3](#_gjdgxs)

[2. ФОРМУЛИРОВАКА ЗАДАЧИ 3](#_1fob9te)

[3. Технология локального подключения фреймворка 3](#_z337ya)

[4. Таблица соответствия переменных и методов, используемых в *web*-приложениях 3](#_3j2qqm3)

[5. Пункт 1 3](#_3znysh7)

[5.1. Код web-приложения 4](#_2et92p0)

[5.2. Результат отображения в браузере 4](#_tyjcwt)

[6. пункт 2 4](#_3dy6vkm)

[6.1. Код web-приложения 4](#_1t3h5sf)

[6.2. Результат отображения в браузере 4](#_4d34og8)

[7. Пункт 3 4](#_2s8eyo1)

[7.1. Код web-приложения 4](#_17dp8vu)

[7.2. Результат отображения в браузере 4](#_3rdcrjn)

[7.3. Сеть петри для 3 пункта 4](#_26in1rg)

[8. ПУНКТ 4 4](#_lnxbz9)

[8.1. Код web-приложения 4](#_35nkun2)

[8.2. Результат отображения в браузере 4](#_1ksv4uv)

[8.3. Сеть петри для 4 пункта 4](#_44sinio)

[9. вывод по работе 4](#_2jxsxqh)

1. **Цель работы**

Освоить навыки создания простых локальных одностраничных web-приложений под управлением Фреймворка Vue.js на языке JavaScript.

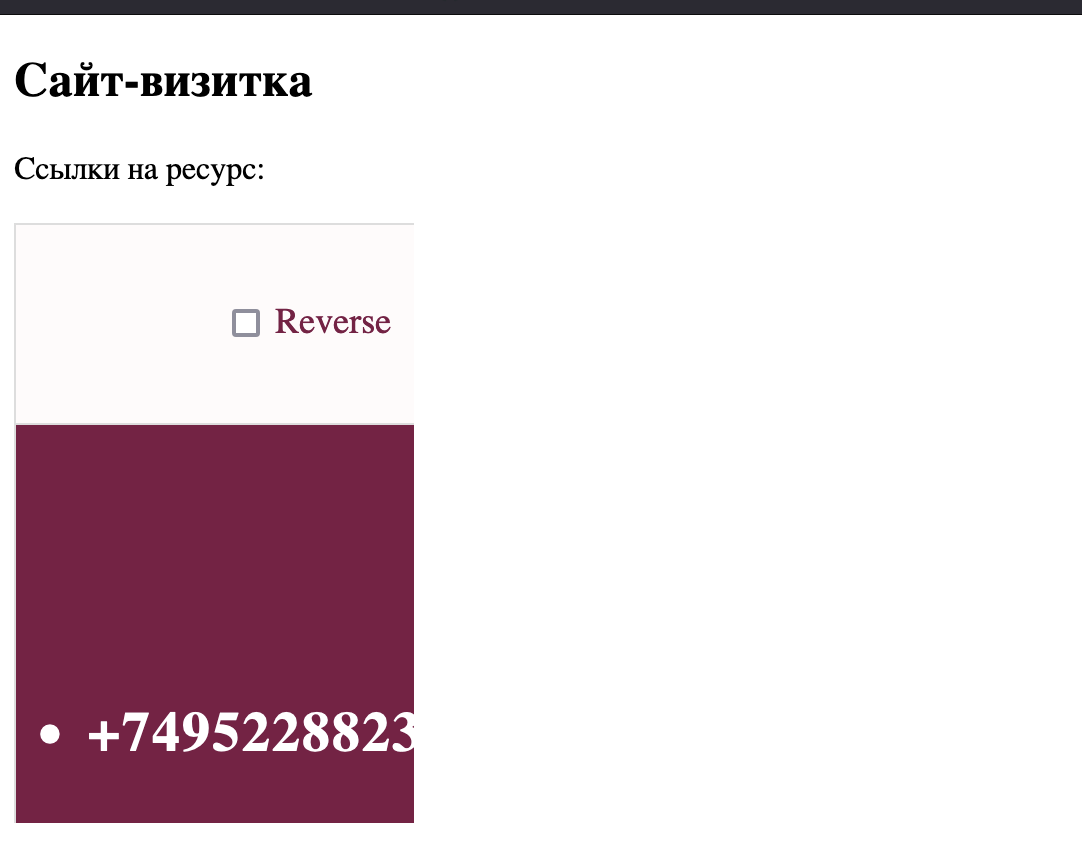
1. **ФОРМУЛИРОВАКА ЗАДАЧИ**

Разработать четыре простых локальных одностраничных web-приложение под управлением фреймворка Vue.js на языке JavaScript в соответствии с указаниями к заданию.

1. **Пункт 1 (Web-страница 1)**
   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>Параметрическая вставка</title>  <script src="https://cdn.jsdelivr.net/npm/vue@2.6.14"></script>  </head>  <style>  .iframe {  left:30px;  top: 20px;  }  </style>  <body>  <div id="app">  <h2>Сайт-визитка</h2>  <p>Ссылки на ресурс:</p>  <iframe v-bind:src="url" v-bind:title="title" v-bind:frameborder="border" v-bind:scrolling="scroll" v-bind:width="width" v-bind:height="height" v-bind:class="move">{{ Name }}</iframe><br>  </div>  <script>  new Vue({  el: '#app',  data: {  url: '../task\_2/index.html',  title: 'Сайт визитка',  Name: 'Ссылка на ресурс',  target: '\_blank',  border: 0,  scroll: "no",  width:200,  height: 300,  move: "iframe"  }  });  </script>  </body>  </html> |

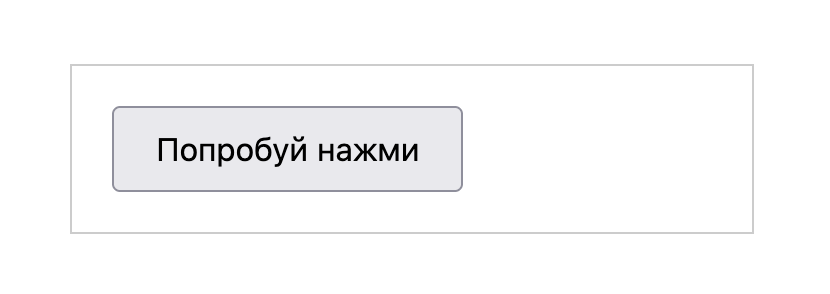
* 1. **Результат отображения в браузере**



1. Результат выполнения пункта 1
2. **пункт 2 (Web-старница 2)**
   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>Vue.js Random Alignment</title>  <script src="https://cdn.jsdelivr.net/npm/vue@3/dist/vue.global.js"></script>  <style>  body {  margin: 0;  padding: 0;  display: flex;  justify-content: center;  align-items: center;  height: 100vh; /\* Центрирование страницы \*/  }  button {  padding: 10px 20px;  font-size: 16px;  cursor: pointer;  }  .message {  font-size: 24px;  font-weight: bold;  }  </style>  </head>  <body>  <div id="app">  <!-- Контейнер -->  <div  :style="{ textAlign: alignment }"  style="width: 300px; border: 1px solid #ccc; padding: 20px;"  >  <!-- Кнопка или сообщение -->  <button  v-if="showButton"  @mouseover="changeAlignment"  @click="hideButton"  >  Попробуй нажми  </button>  <div  v-else  class="message"  >  Удача!  </div>  </div>  </div>  <script>  const App = {  data() {  return {  alignment: "left", // Начальное выравнивание  showButton: true, // Видимость кнопки  };  },  methods: {  // Случайный выбор выравнивания  changeAlignment() {  const alignments = ["left", "center", "right"];  this.alignment = alignments[Math.floor(Math.random() \* alignments.length)];  },  // Скрытие кнопки и вывод сообщения  hideButton() {  this.showButton = false;  }  }  };  // Инициализация Vue-приложения  Vue.createApp(App).mount("#app");  </script>  </body>  </html> |

* 1. **Результат отображения в браузере**



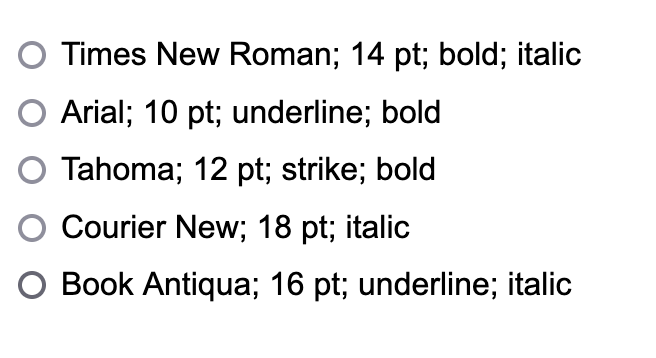
1. Результат выполнения пункта 2



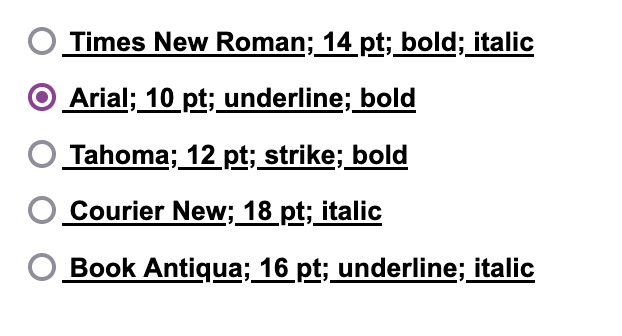
1. Результат выполнения пункта 2
2. **Пункт 3** 
   1. **Код web-приложения**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Vue.js Radio Buttons Global Style</title>  <script src="https://cdn.jsdelivr.net/npm/vue@3/dist/vue.global.js"></script>  <style>  body {  font-family: Arial, sans-serif;  padding: 20px;  }  .radio-group {  margin: 20px 0;  }  label {  display: block;  margin-bottom: 10px;  cursor: pointer;  }  </style>  </head>  <body>  <div id="app">  <div class="radio-group">  <!-- Радиокнопки с описанием -->  <label  v-for="(option, index) in options"  :key="index"  :style="globalStyle"  >  <input  type="radio"  :value="index"  v-model="selectedStyle"  />  {{ option.label }}  </label>  </div>  </div>  <script>  const App = {  data() {  return {  // Список радиокнопок с соответствующими стилями  options: [  { label: "Times New Roman; 14 pt; bold; italic", style: { fontFamily: "Times New Roman", fontSize: "14pt", fontWeight: "bold", fontStyle: "italic" } },  { label: "Arial; 10 pt; underline; bold", style: { fontFamily: "Arial", fontSize: "10pt", fontWeight: "bold", textDecoration: "underline" } },  { label: "Tahoma; 12 pt; strike; bold", style: { fontFamily: "Tahoma", fontSize: "12pt", fontWeight: "bold", textDecoration: "line-through" } },  { label: "Courier New; 18 pt; italic", style: { fontFamily: "Courier New", fontSize: "18pt", fontStyle: "italic" } },  { label: "Book Antiqua; 16 pt; underline; italic", style: { fontFamily: "Book Antiqua", fontSize: "16pt", fontStyle: "italic", textDecoration: "underline" } }  ],  selectedStyle: null // Индекс выбранного стиля  };  },  computed: {  // Применение стиля ко всем ярлыкам  globalStyle() {  return this.selectedStyle !== null  ? this.options[this.selectedStyle].style  : {};  }  }  };  Vue.createApp(App).mount("#app");  </script>  </body>  </html> |

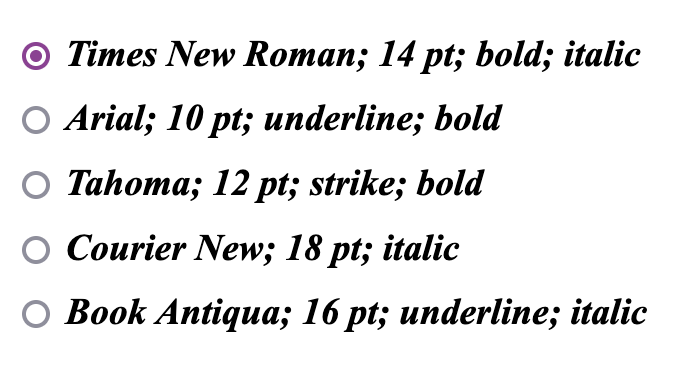
* 1. **Результат отображения в браузере**



1. Результат выполнения пункта 3

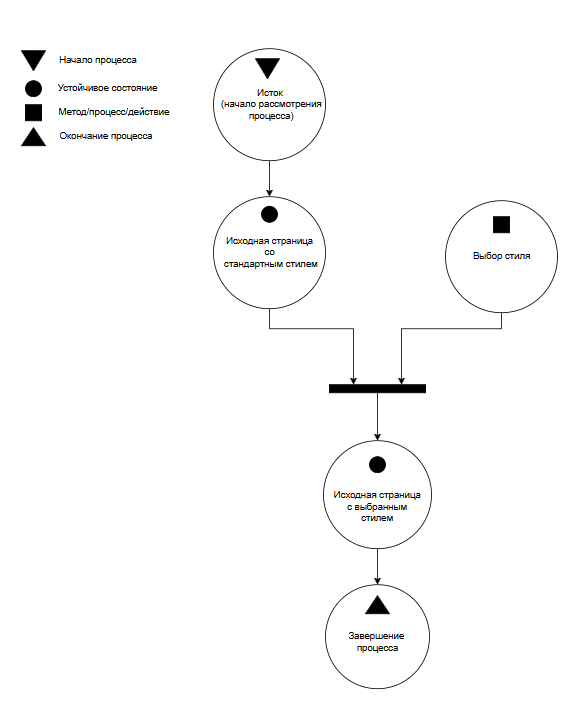


1. Результат выполнения пункта 3



1. Результат выполнения пункта 3

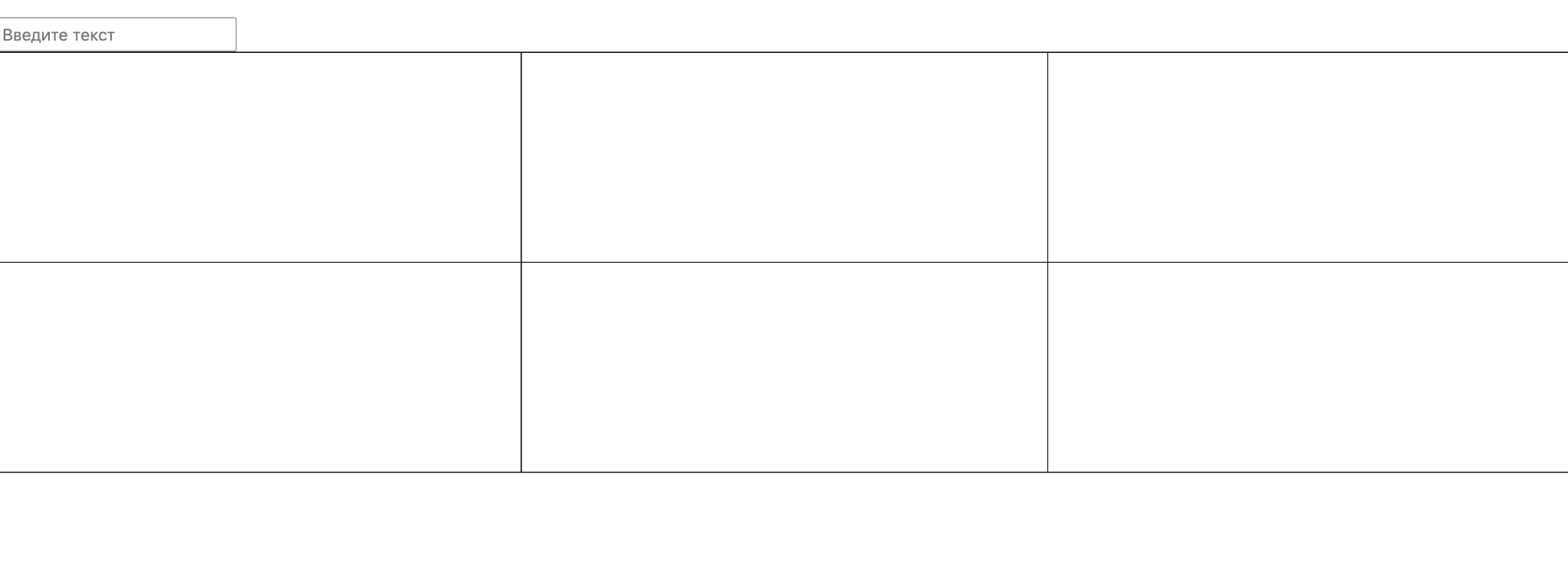
* 1. **Сети Петри для 3 пункта**



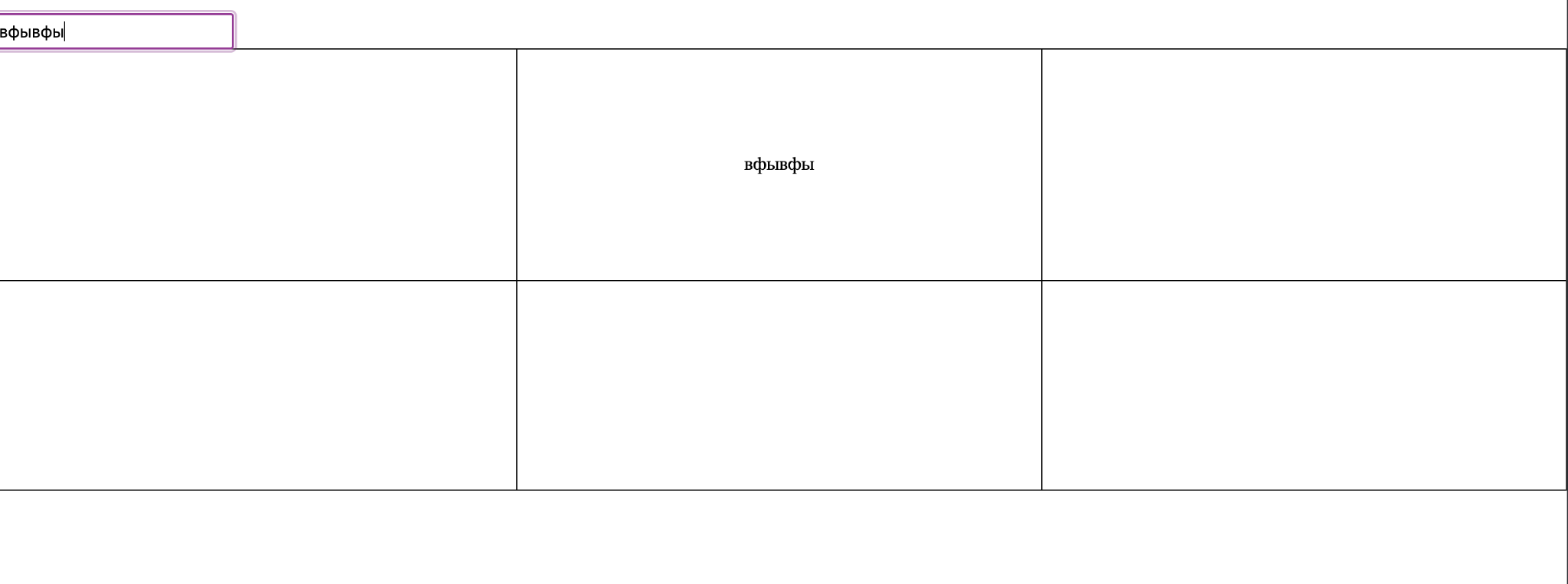
1. Сети Петри для 3 пункта
2. **ПУНКТ 4** 
   1. **Код web-приложения**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Vue.js Table Navigation</title>  <script src="https://cdn.jsdelivr.net/npm/vue@3/dist/vue.global.js"></script>  <style>  body {  display: flex;  height: 100vh;  margin: 0;  align-items: center;  justify-content: center;  }  input {  margin-right: 20px;  padding: 5px;  font-size: 16px;  }  table {  border-collapse: collapse;  width: 100%;  height: 100%;  table-layout: fixed;  }  td {  border: 1px solid black;  text-align: center;  vertical-align: middle;  font-size: 18px;  position: relative;  padding: 100px;  }  </style>  </head>  <body>  <div id="app">  <!-- Поле для ввода текста -->  <input  v-model="text"  type="text"  placeholder="Введите текст"  />  <!-- Таблица -->  <table>  <tr v-for="(row, rowIndex) in tableRows" :key="rowIndex">  <td  v-for="(cell, colIndex) in row"  :key="colIndex"  :style="getCellStyle(rowIndex, colIndex)"  >  <!-- Отображение текста в текущей ячейке -->  <span v-if="currentRow === rowIndex && currentCol === colIndex">{{ text }}</span>  </td>  </tr>  </table>  </div>  <script>  const App = {  data() {  return {  text: "", // Текст из input  tableRows: Array(2).fill(Array(3).fill(null)), // Таблица 2x3  currentRow: 0, // Текущая строка  currentCol: 1, // Текущий столбец (центральная верхняя ячейка)  horizontalAlignments: ["left", "center", "right"], // Горизонтальные выравнивания  verticalAlignments: ["top", "middle", "bottom"], // Вертикальные выравнивания  hIndex: 1, // Индекс текущего горизонтального выравнивания (начинается с center)  vIndex: 1, // Индекс текущего вертикального выравнивания (начинается с middle)  };  },  methods: {  // Получение стиля для текущей ячейки  getCellStyle(row, col) {  if (this.currentRow === row && this.currentCol === col) {  return {  textAlign: this.horizontalAlignments[this.hIndex],  verticalAlign: this.verticalAlignments[this.vIndex],  };  }  return {};  },  // Обработка нажатия клавиш  handleKey(event) {  const key = event.key;  if (key === "ArrowLeft") this.moveHorizontal(-1); // ←  else if (key === "ArrowRight") this.moveHorizontal(1); // →  else if (key === "ArrowUp") this.moveVertical(-1); // ↑  else if (key === "ArrowDown") this.moveVertical(1); // ↓  },  // Перемещение по горизонтали  moveHorizontal(direction) {  this.hIndex += direction;  if (this.hIndex < 0) {  this.hIndex = this.horizontalAlignments.length - 1;  if (this.currentCol > 0) this.currentCol--; // Влево по ячейке  } else if (this.hIndex >= this.horizontalAlignments.length) {  this.hIndex = 0;  if (this.currentCol < 2) this.currentCol++; // Вправо по ячейке  }  },  // Перемещение по вертикали  moveVertical(direction) {  this.vIndex += direction;  if (this.vIndex < 0) {  this.vIndex = this.verticalAlignments.length - 1;  if (this.currentRow > 0) this.currentRow--; // Вверх по ячейке  } else if (this.vIndex >= this.verticalAlignments.length) {  this.vIndex = 0;  if (this.currentRow < 1) this.currentRow++; // Вниз по ячейке  }  },  },  mounted() {  // Добавление слушателя событий клавиатуры  window.addEventListener("keydown", this.handleKey);  },  beforeUnmount() {  // Удаление слушателя при уничтожении компонента  window.removeEventListener("keydown", this.handleKey);  },  };  Vue.createApp(App).mount("#app");  </script>  </body>  </html> |

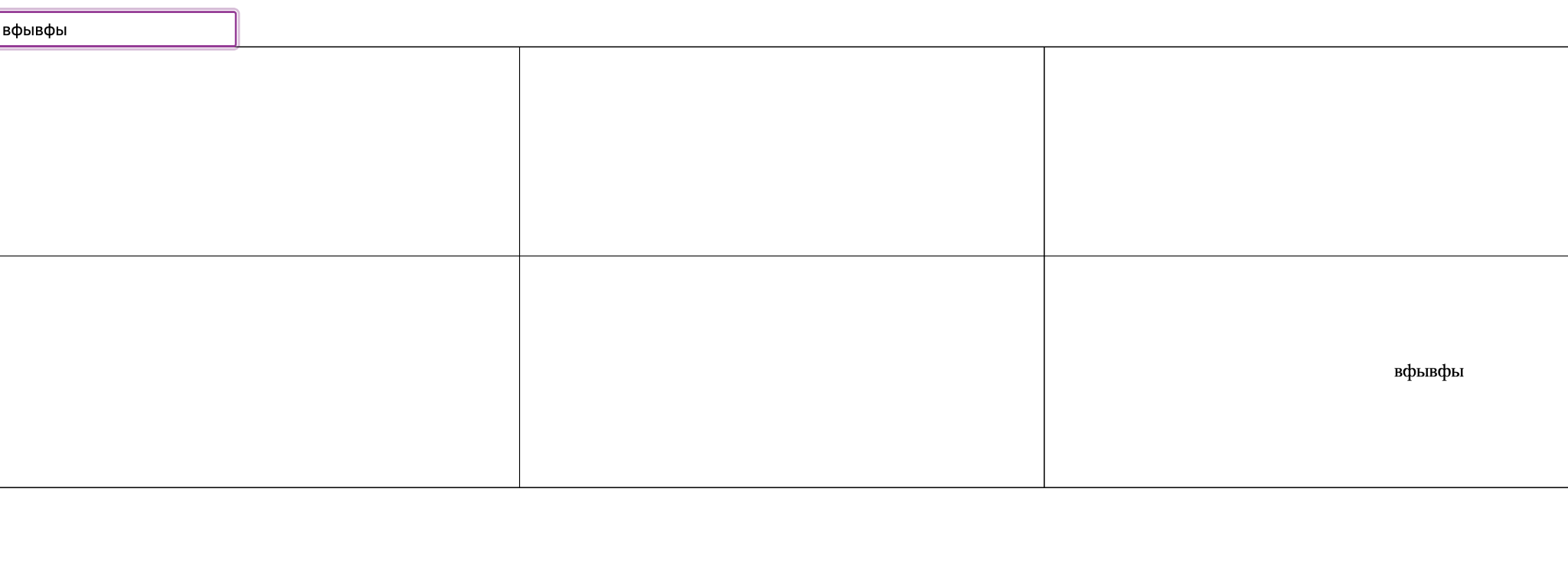
* 1. **Результат отображения в браузере**



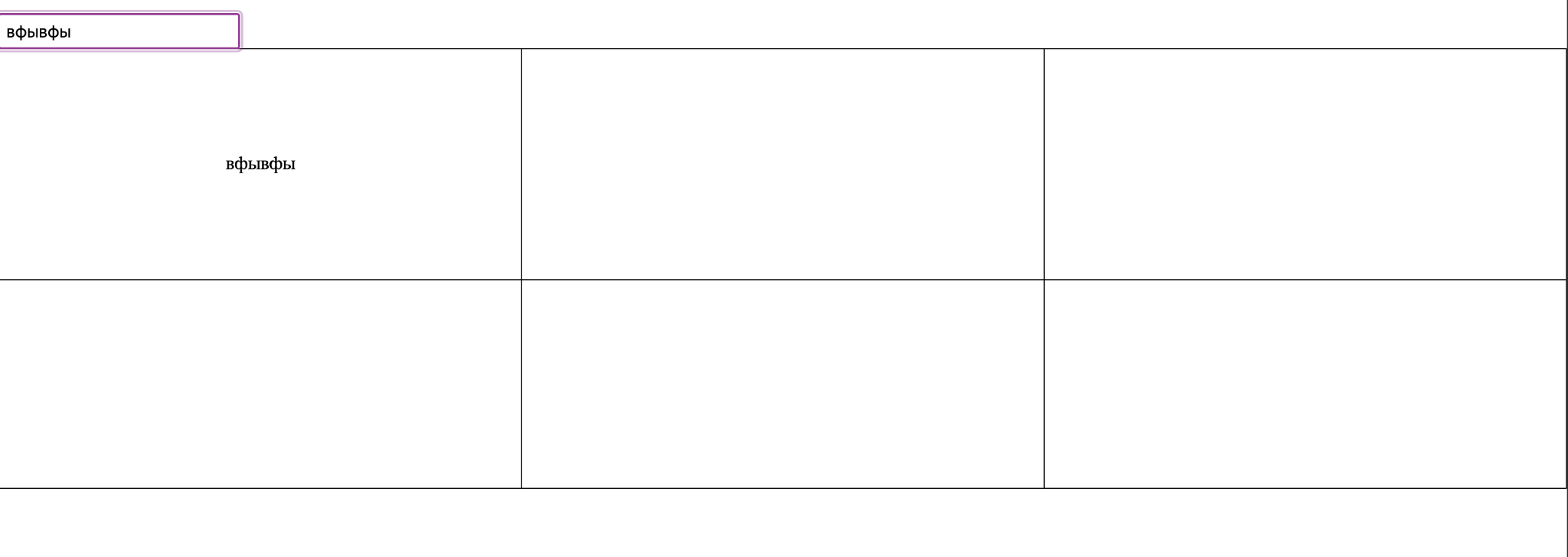
1. Результат выполнения пункта 4

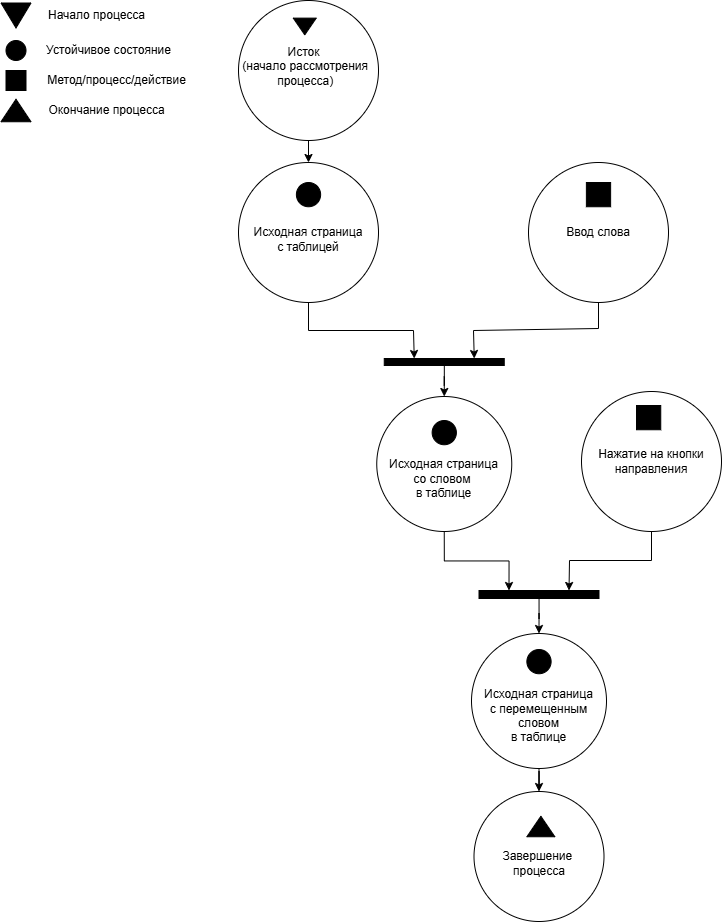


1. Результат выполнения пункта 4



1. Результат выполнения пункта 4

****

1. Результат выполнения пункта 4
   1. **Сети Петри для 4 пункта**
2. 
3. Сети Петри 4 пункта
4. **вывод по работе**

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