|  |  |
| --- | --- |
| **Российский университет транспорта (МИИТ)**  **Институт транспортной техники и систем управления**  **Кафедра «Управление и защита информации»** | |
| **Задание №4**  **по теме «Технология локального подключения фреймворков»**  **по дисциплине «Web-программирование»** | |
|  | Выполнил:  Студент группы ТКИ-542  Пономарев А.Д.  Дроздов А.Д.  Проверил:  Доцент кафедры УиЗИ, к.т.н., с.н.с  Сафронов А.И. |
| Москва 2024 | |

Оглавление

[1. Цель работы 3](#_Toc181105810)

[2. ФОРМУЛИРОВАКА ЗАДАЧИ 3](#_Toc181105811)

[3. Пункт 1 (Web-страница 1) 4](#_Toc181105812)

[3.1. Формулировка задачи 4](#_Toc181105813)

[3.2. Код web-приложения 4](#_Toc181105814)

[3.3. Результат отображения в браузере 10](#_Toc181105815)

[3.4. Схема Сетей Петри 11](#_Toc181105816)

[4. Пункт 1 (Web-страница 1) 12](#_Toc181105817)

[4.1. Формулировка задачи 12](#_Toc181105818)

[4.2. Код web-приложения 12](#_Toc181105819)

[4.3. Результат отображения в браузере 14](#_Toc181105820)

[4.4. Схема Сетей Петри 15](#_Toc181105821)

[5. пункт 3 (Web-старница 3) 16](#_Toc181105822)

[5.1. Формулировка задачи 16](#_Toc181105823)

[5.2. Код web-приложения 16](#_Toc181105824)

[5.3. Результат отображения в браузере 17](#_Toc181105825)

[5.4. Схема Сетей Петри 18](#_Toc181105826)

[6. вывод по работе 19](#_Toc181105827)

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. ФОРМУЛИРОВАКА 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.*

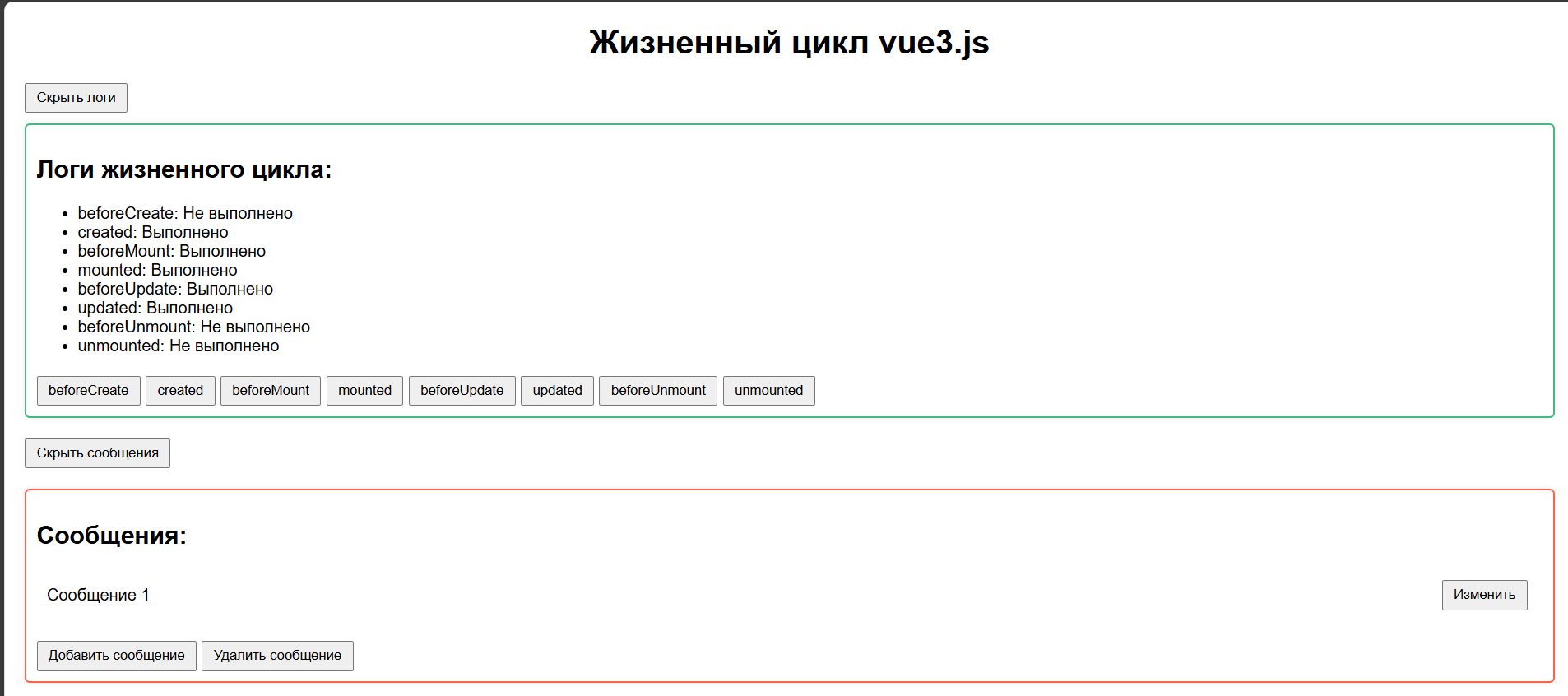
1. Пункт 1 (Web-страница 1)
   1. Формулировка задачи

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).

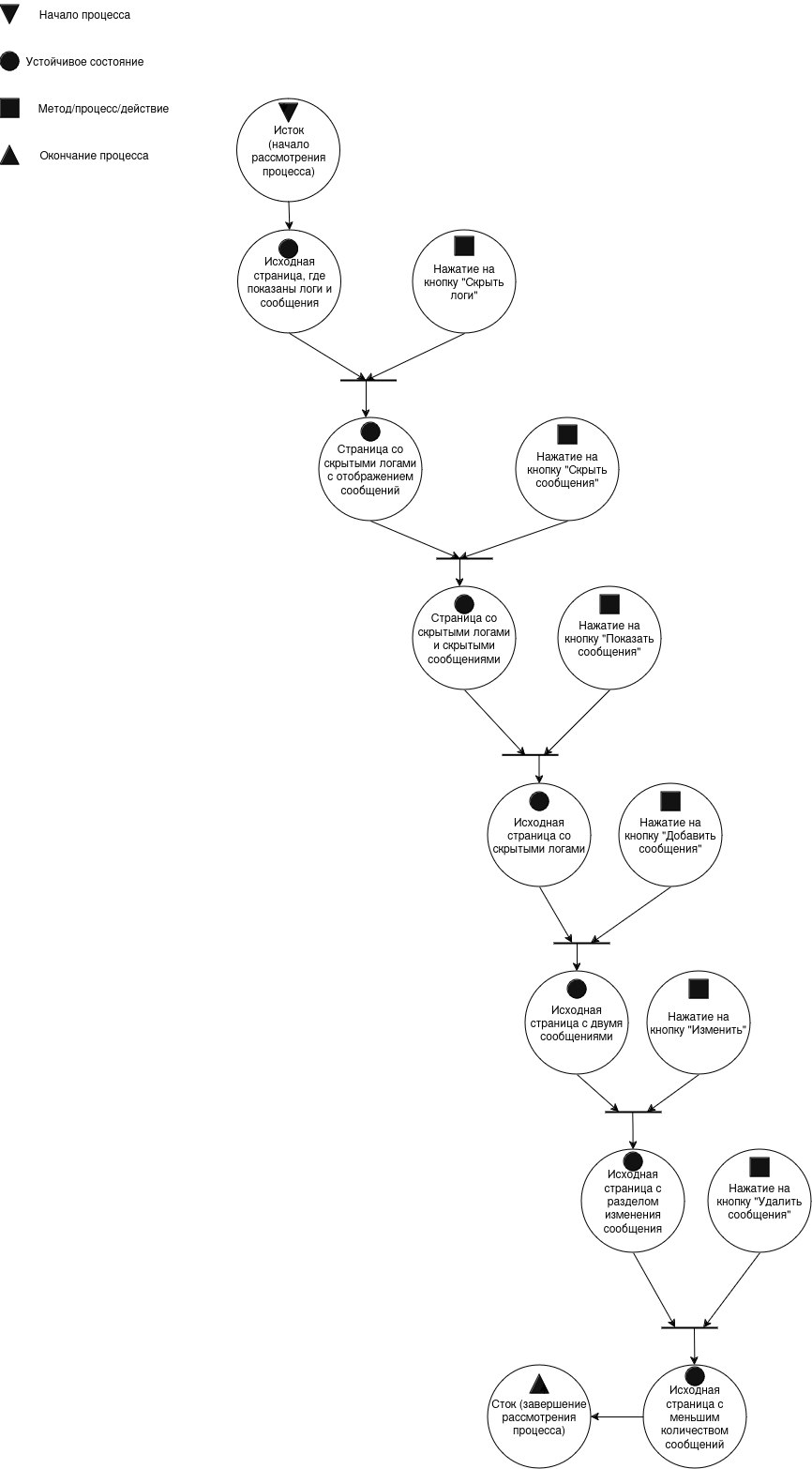
* 1. Код web-приложения

|  |
| --- |
| <!DOCTYPE html>  <html lang=**"ru"**>  <head>  <meta charset=**"UTF-8"**>  <title>**Жизненный цикл**</title>  <script src=**"https://unpkg.com/vue@3/dist/vue.global.prod.js"**></script>  <style>  **body {**  **font-family: Arial, sans-serif;**  **margin: 20px;**  **}**  **.hook-log {**  **margin-bottom: 20px;**  **padding: 10px;**  **border: 2px solid #42b983;**  **border-radius: 5px;**  **box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);**  **}**  **.messages {**  **margin-top: 20px;**  **padding: 10px;**  **border: 2px solid #ff6347;**  **border-radius: 5px;**  **box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);**  **}**  **.message {**  **padding: 10px;**  **border-bottom: 1px solid #eee;**  **display: flex;**  **justify-content: space-between;**  **align-items: center;**  **}**  **.message:last-child {**  **border-bottom: none;**  **}**  **button {**  **margin-right: 5px;**  **padding: 5px 10px;**  **}**  **.controls {**  **margin-top: 20px;**  **}**  **.edit-input {**  **margin-top: 10px;**  **}**  **h1 {**  **text-align: center;**  **}**  **.log-toggle {**  **margin-bottom: 10px;**  **}**  **.message-toggle {**  **margin-bottom: 10px;**  **}**  </style>  </head>  <body>  <div id=**"app"**>  <h1>**Жизненный цикл vue3.js**</h1>    <div class=**"log-toggle"**>  <button @click=**"toggleLogs"**>  **{{ showLogs ? 'Скрыть логи' : 'Показать логи' }}**  </button>  </div>  <div class=**"hook-log"** v-if=**"showLogs"**>  <h2>**Логи жизненного цикла:**</h2>  <ul>  <li>**beforeCreate: {{ logs.beforeCreate }}**</li>  <li>**created: {{ logs.created }}**</li>  <li>**beforeMount: {{ logs.beforeMount }}**</li>  <li>**mounted: {{ logs.mounted }}**</li>  <li>**beforeUpdate: {{ logs.beforeUpdate }}**</li>  <li>**updated: {{ logs.updated }}**</li>  <li>**beforeUnmount: {{ logs.beforeUnmount }}**</li>  <li>**unmounted: {{ logs.unmounted }}**</li>  </ul>  <div class=**"controls"**>  <button @click=**"triggerBeforeCreate"**>**beforeCreate**</button>  <button @click=**"triggerCreated"**>**created**</button>  <button @click=**"triggerBeforeMount"**>**beforeMount**</button>  <button @click=**"triggerMounted"**>**mounted**</button>  <button @click=**"triggerBeforeUpdate"**>**beforeUpdate**</button>  <button @click=**"triggerUpdated"**>**updated**</button>  <button @click=**"triggerBeforeUnmount"**>**beforeUnmount**</button>  <button @click=**"triggerUnmounted"**>**unmounted**</button>  </div>  </div>  <div class=**"message-toggle"**>  <button @click=**"toggleMessages"**>  **{{ showMessages ? 'Скрыть сообщения' : 'Показать сообщения' }}**  </button>  </div>  <div class=**"messages"** v-if=**"showMessages"**>  <h2>**Сообщения:**</h2>  <div v-if=**"messages.length === 0"**>**Нет сообщений.**</div>  <div v-else>  <div class=**"message"** v-for=**"(message, index) in messages"** :key=**"index"**>  <span>**{{ message }}**</span>  <button @click=**"changeMessage(index)"**>**Изменить**</button>  </div>  </div>  <div class=**"controls"**>  <button @click=**"addMessage"**>**Добавить сообщение**</button>  <button @click=**"removeMessage"**>**Удалить сообщение**</button>  </div>  </div>  <div v-if=**"isEditing"** class=**"edit-input"**>  <h3>**Изменить сообщение {{ editingMessage }}**</h3>  <input v-model=**"newMessageText"** placeholder=**"Новое сообщение"** />  <button @click=**"confirmChangeMessage"**>**Изменить**</button>  <button @click=**"cancelChangeMessage"**>**Отменить измения**</button>  </div>  </div>  <script>  **if** **(typeof** Vue **===** 'undefined'**)** **{**  console.error**(**'Vue.js не загружен. Проверьте подключение CDN.'**);**  **}** **else** **{**  **const** **{** createApp**,** reactive **}** **=** Vue**;**  **const** app **=** createApp**({**  data**()** **{**  **return** **{**  logs**:** **{**  beforeCreate**:** 'Не выполнено'**,**  created**:** 'Не выполнено'**,**  beforeMount**:** 'Не выполнено'**,**  mounted**:** 'Не выполнено'**,**  beforeUpdate**:** 'Не выполнено'**,**  updated**:** 'Не выполнено'**,**  beforeUnmount**:** 'Не выполнено'**,**  unmounted**:** 'Не выполнено'  **},**  messages**:** **[],**  isEditing**:** **false,**  editingIndex**:** **null,**  newMessageText**:** ''**,**  showLogs**:** **true,**  showMessages**:** **true**  **}**  **},**  methods**:** **{**  addMessage**()** **{**  **const** newMessage **=** `Сообщение $**{**this.messages.length **+** 1**}**`**;**  this.messages.push**(**newMessage**);**  console.log**(**`Добавлено**:** $**{**newMessage**}**`**);**  **},**  removeMessage**()** **{**  **if** **(**this.messages.length **>** 0**)** **{**  **const** removed **=** this.messages.pop**();**  console.log**(**`Удалено**:** $**{**removed**}**`**);**  **}** **else** **{**  console.log**(**'Нет сообщений для удаления.'**);**  **}**  **},**  changeMessage**(**index**)** **{**  this.editingIndex **=** index**;**  this.newMessageText **=** this.messages**[**index**];**  this.isEditing **=** **true;**  console.log**(**`Редактирование сообщения $**{**index **+** 1**}**`**);**  **},**  confirmChangeMessage**()** **{**  **if** **(**this.newMessageText.trim**()** **!==** ''**)** **{**  **const** oldMessage **=** this.messages**[**this.editingIndex**];**  this.messages**[**this.editingIndex**]** **=** this.newMessageText.trim**();**  console.log**(**`Сообщение $**{**this.editingIndex **+** 1**}** изменено с "${oldMessage}" на "${this.newMessageText.trim()}"`**);**  this.isEditing **=** **false;**  this.editingIndex **=** **null;**  this.newMessageText **=** ''**;**  **}** **else** **{**  alert**(**'Текст сообщения не может быть пустым.'**);**  **}**  **},**  cancelChangeMessage**()** **{**  this.isEditing **=** **false;**  this.editingIndex **=** **null;**  this.newMessageText **=** ''**;**  console.log**(**'Редактирование сообщения отменено.'**);**  **},**  unmountApp**()** **{**  app.unmount**();**  console.log**(**'Приложение размонтировано.'**);**  **},**  toggleLogs**()** **{**  this.showLogs **=** **!**this.showLogs**;**  **},**  toggleMessages**()** **{**  this.showMessages **=** **!**this.showMessages**;**  **},**  triggerBeforeCreate**()** **{**  this.logs.beforeCreate **=** 'Выполнено'**;**  console.log**(**'beforeCreate вызван'**);**  **},**  triggerCreated**()** **{**  this.logs.created **=** 'Выполнено'**;**  console.log**(**'created вызван'**);**  **},**  triggerBeforeMount**()** **{**  this.logs.beforeMount **=** 'Выполнено'**;**  console.log**(**'beforeMount вызван'**);**  **},**  triggerMounted**()** **{**  this.logs.mounted **=** 'Выполнено'**;**  console.log**(**'mounted вызван'**);**  **},**  triggerBeforeUpdate**()** **{**  this.logs.beforeUpdate **=** 'Выполнено'**;**  console.log**(**'beforeUpdate вызван'**);**  **},**  triggerUpdated**()** **{**  this.logs.updated **=** 'Выполнено'**;**  console.log**(**'updated вызван'**);**  **},**  triggerBeforeUnmount**()** **{**  this.logs.beforeUnmount **=** 'Выполнено'**;**  console.log**(**'beforeUnmount вызван'**);**  **},**  triggerUnmounted**()** **{**  this.logs.unmounted **=** 'Выполнено'**;**  console.log**(**'unmounted вызван'**);**  **}**  **},**  created**()** **{**  console.log**(**'created вызван'**);**  this.messages.push**(**'Сообщение 1'**);**  this.logs.created **=** 'Выполнено'**;**  **},**  beforeCreate**()** **{**  console.log**(**'beforeCreate вызван'**);**  this.logs.beforeCreate **=** 'Выполнено'**;**  **},**  beforeMount**()** **{**  console.log**(**'beforeMount вызван'**);**  this.logs.beforeMount **=** 'Выполнено'**;**  **},**  mounted**()** **{**  console.log**(**'mounted вызван'**);**  this.logs.mounted **=** 'Выполнено'**;**  **},**  beforeUpdate**()** **{**  console.log**(**'beforeUpdate вызван'**);**  this.logs.beforeUpdate **=** 'Выполнено'**;**  **},**  updated**()** **{**  console.log**(**'updated вызван'**);**  this.logs.updated **=** 'Выполнено'**;**  **},**  beforeUnmount**()** **{**  console.log**(**'beforeUnmount вызван'**);**  this.logs.beforeUnmount **=** 'Выполнено'**;**  **},**  unmounted**()** **{**  console.log**(**'unmounted вызван'**);**  this.logs.unmounted **=** 'Выполнено'**;**  **}**  **});**  app.mount**(**'#app'**);**  **}**  </script>  </body>  </html |

* 1. Результат отображения в браузере



1. Результат выполнения пункта 1
   1. Схема Сетей Петри



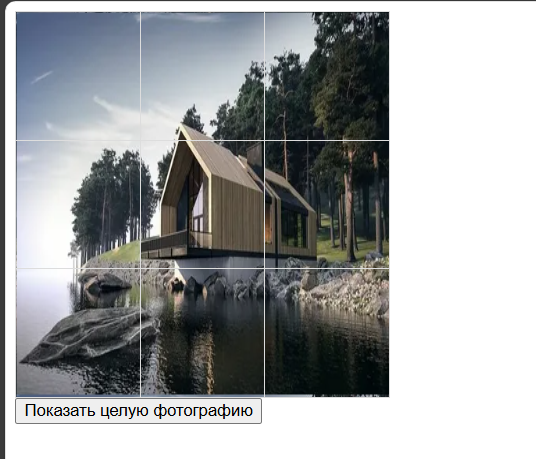
1. Схема Петри пункта 1
2. Пункт 1 (Web-страница 1)
   1. Формулировка задачи

**Вариант 5.** Некоторое изображение нарезано стык-в-стык на 9 частей таким образом, чтобы оно фрагментарно размещалось в необрамлённой таблице <*table*> размерности [*3 x 3*] при использовании директивы *v-for*.

* 1. Код web-приложения

|  |
| --- |
| <!DOCTYPE html>  <html lang="ru">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Нарезанное изображение в таблице 3x3</title>  <script src="https://cdn.jsdelivr.net/npm/vue@2"></script>  <style>  table {  border-collapse: collapse;  width: 300px;  height: 300px;  }  td {  width: 100px;  height: 100px;  border: 1px solid #ddd;  background-image: url('D:/5 курс/1 семестр/web/4/i.jpg');  background-size: 300px 300px;  }  .whole-image {  width: 300px;  height: 300px;  background-image: url('D:/5 курс/1 семестр/web/4/i.jpg');  background-size: 300px 300px;  background-position: 0 0;  }  </style>  </head>  <body>  <div id="app">  <div v-if="!sliced" class="whole-image"></div>  <table v-else>  <tr v-for="row in 3" :key="row">  <td  v-for="col in 3"  :key="col"  :style="getSliceStyle((row - 1) \* 3 + (col - 1))">  </td>  </tr>  </table>  <button @click="toggleSliced">  {{ sliced ? 'Показать целую фотографию' : 'Разделить фотографию' }}  </button>  </div>  <script>  new Vue({  el: '#app',  data: {  sliced: false  },  methods: {  getSliceStyle(index) {  const x = (index % 3) \* -100;  const y = Math.floor(index / 3) \* -100;  return { backgroundPosition: `${x}px ${y}px` };  },  toggleSliced() {  this.sliced = !this.sliced;  }  }  });  </script>  </body>  </html> |

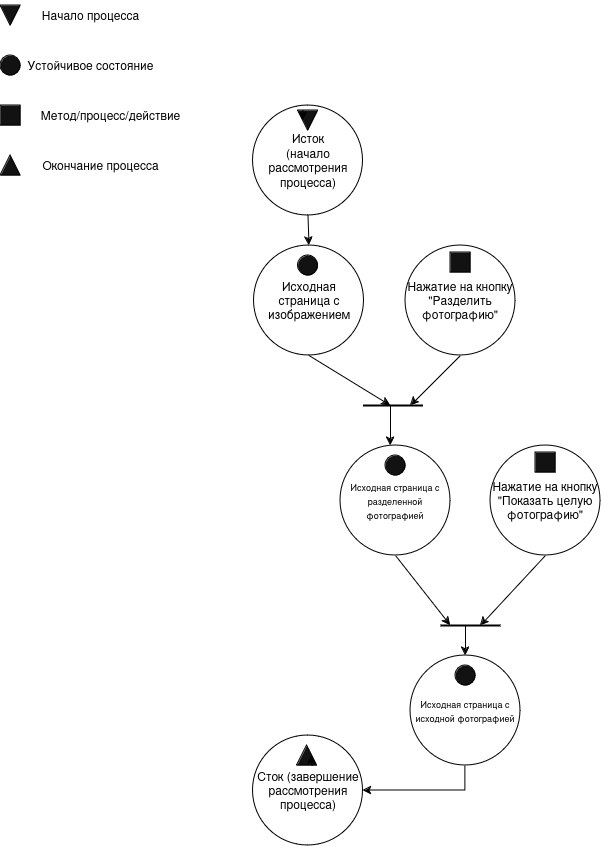
* 1. Результат отображения в браузере



1. Результат выполнения пункта 2



1. Результат выполнения пункта 2
   1. Схема Сетей Петри



1. Схема Петри пункта 2
2. пункт 3 (Web-старница 3)
   1. Формулировка задачи

**Вариант 5.**Список <*select*> наполняется последовательно пунктами, наименование которых вводятся в <*input*> и подтверждается нажатием на «*Enter*». Выставляется ограничение на ввод – запрещено вводить более 10 символов. Под это условие подобрать соответствующую, фиксированную ширину <*select*>. Учесть при подборе ширины <*select*>, что могут вводиться и все заглавные символы.

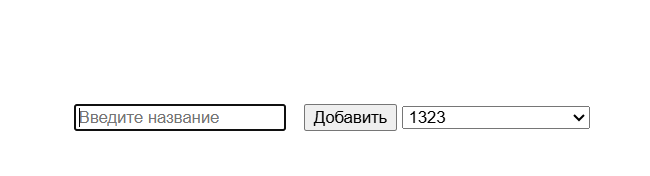
* 1. Код web-приложения

|  |
| --- |
| <!DOCTYPE html>  <html lang="ru">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Наблюдатели в Vue</title>  <script src="https://cdn.jsdelivr.net/npm/vue@2"></script>  <style>  body {  font-family: Arial, sans-serif;  display: flex;  justify-content: center;  align-items: center;  height: 100vh;  margin: 0;  }  #app {  text-align: center;  }  select {  width: 150px;  }  input {  margin-right: 10px;  }  </style>  </head>  <body>  <div id="app">  <input type="text" v-model="inputValue" @keydown.enter="addItem" maxlength="10" placeholder="Введите название" />  <button @click="addItem">Добавить</button>  <select>  <option v-for="item in items" :key="item">{{ item }}</option>  </select>  </div>  <script>  new Vue({  el: '#app',  data: {  inputValue: '',  items: []  },  methods: {  addItem() {  if (this.inputValue.trim() === '' || this.items.length >= 10) { return; }  this.items.push(this.inputValue);  this.inputValue = '';  }  },  watch: {  items(newItems) {  console.log('Список товаров изменен:', newItems);  }  }  });  </script>  </body>  </html> |

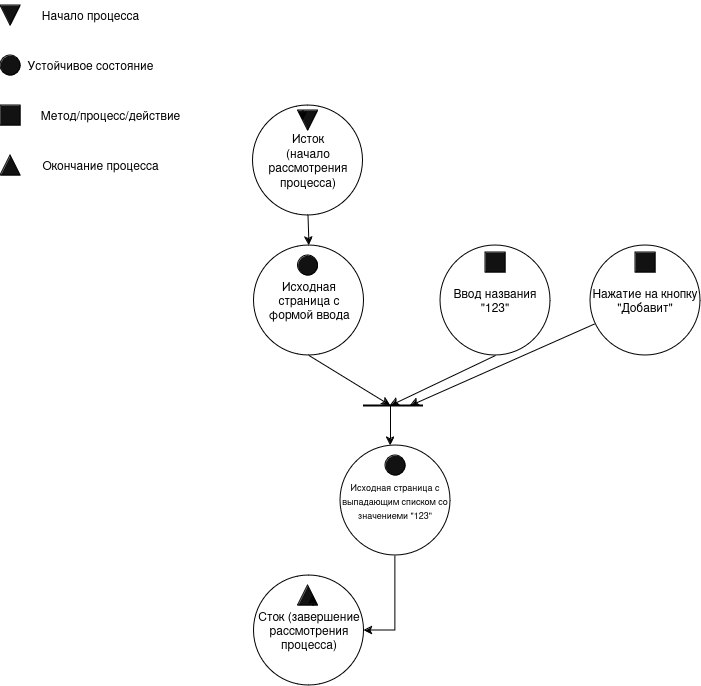
* 1. Результат отображения в браузере



1. Результат выполнения пункта 3



1. Результат выполнения пункта 3
   1. Схема Сетей Петри



1. Схема Петри пункта 3
2. вывод по работе

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