Министерство транспорта Российской Федерации

Федеральное государственное автономное

образовательное учреждение высшего образования

«Российский университет транспорта (МИИТ)»

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа №4

по дисциплине:

«Web-программирование»

на тему:  
«Последовательный и разветвляющийся вычислительные процессы»

Выполнил: ст. гр. ТКИ-541

Черкас Н.В., Родин К.П.

Вариант №13

Проверил: к.т.н., доцент Сафронов А.И.

Москва – 2025 г.

# **1. Цель работы**

Целью данной лабораторной работы является приобретение практических навыков разработки простых локальных *web*-приложений (*LSPWA*) под управлением фреймворка *Vue.js* на языке *JavaScript*.

# **2. Формулировка задачи**

Разработать четыре простых локальных одностраничных *web*-приложение (*LSPWA*) под управлением фреймворка *Vue.js* на языке *JavaScript* в соответствии с указаниями вариантов индивидуального задания (покрываются три раздела последовательного вычислительного процесса и один раздел разветвляющегося вычислительного процесса). Допустимо выполнение в синтаксисе как *Vue.js 2.x*, так и *Vue.js 3.x*.

*Базовая структура отчёта по работе:*

1. Цель работы.

2. Формулировка задачи.

3. Технология **локального** *(****LSPWA****)* подключения фреймворка *Vue.js.*

4. Таблица соответствия переменных и методов, используемых в *web*-приложениях (для случаев, когда переменные названы не значимо).

5. Содержательная часть по **параметрической вставке** (*v-bind;* код *web*-приложения + отображение в браузере (с учётом изменяющихся состояний) + схема ТП согласно нотации, определённой рабочим коллективом).

6. Содержательная часть по **параметрической связке** (*v-model*; код *web*-приложения + отображение в браузере (с учётом изменяющихся состояний) + схема ТП согласно нотации, определённой рабочим коллективом).

7. Содержательная часть по **методам и событиям** (*v-on*; код *web*-приложения + отображение в браузере (с учётом изменяющихся состояний) + схема ТП согласно нотации, определённой рабочим коллективом).

8. Содержательная часть по **условной отрисовке** (*v-if*; код *web*-приложения + отображение в браузере (с учётом изменяющихся состояний) + схема ТП согласно нотации, определённой рабочим коллективом).

9. Вывод.

В каждом разделе учесть ограничения, прописанные в жёлтых клетках таблиц.

**Реализовать:**

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

– в файле с именем *index01\_Familiya\_I\_O1\_Familiya\_I\_O2.html* параметрическую вставку;

– в файле с именем *index02\_Familiya\_I\_O1\_Familiya\_I\_O2.html* параметрическую связку;

– в файле с именем *index03\_Familiya\_I\_O1\_Familiya\_I\_O2.html* работу с методами и событиями;

– в файле с именем *index04\_Familiya\_I\_O1\_Familiya\_I\_O2.html* условную отрисовку.

**Вариант для приложения 4.1:**

Настроить при помощи параметрической подстановки якорный тег <*a*></*a*>. Подготовить две ссылки на один и тот же ресурс (предпочтительно, чтобы это была локально размещённая копия сайта-визитки, созданного в Задании 2). Ссылка передаётся параметрически как значение одного и того же параметра. Создать всплывающую подсказку с наименованием ресурса. Между якорным тегом <*a*></*a*> посредством подстановки в «усатых скобках» заключается имя файла (или ресурса), на который переводит ссылка. Параметрически задавать открытие страницы по ссылке в новой вкладке (*target="\_blank"*).

**Вариант для приложения 4.2:**

Настроить при помощи параметрической подстановки тег нумерованного списка <*ol*></*ol*>. Предусмотреть три позиции в этом списке. Предусмотреть экранные кнопки <*button*><*/button*> «Вперёд» и «Назад» для смены в соответствующем направлении вида нумератора для элементов списка из числа возможных (стандартных, НЕ пользовательских) для нумерованного списка. При выборе первого вида нумераторов не реагировать на нажатие кнопки «Назад», при выборе последнего вида нумераторов не реагировать на нажатие кнопки «Вперёд». Предусмотреть дополнительные экранные кнопки «Больше» и «Меньше», посредством которых возможно менять числовое значение для начала списка. Запретить попадание числовых значений в область отрицательных посредством отсутствия в соответствующий момент реакции на кнопку «Меньше» (кнопка доступна для нажатия, но это не приводит ни к какому результату).

**Вариант для приложения 4.3:**

На *web*-страницу выводится заблокированный (для взаимодействия с пользователем) экранный список, состоящий из пяти элементов *<select></select>*. Все пять пунктов демонстрируются – не менее и не более установленного количества. Оператор управляет выделенной позицией в списке посредством нажатия на экранные кнопки с надписями вида: «Предыдущий» и «Следующий». При выборе последнего элемента кнопка «Следующий» становится недоступной для пользователя. При выборе первого элемента кнопка «Предыдущий» становится недоступной для пользователя.

**Вариант для приложения 4.4:**

Предусмотреть две экранных кнопки, первая из которых последовательно прокручивает шаблоны (*templates*) из Приложения в последовательности «*Author*» */* «*Title*» */* «*Editor*» */* «*Publisher*», после чего становится невидимой и передаёт управление другой экранной кнопке, становящейся видимой. Вторая кнопка, в свою очередь, тоже последовательно прокручивает шаблоны «*Author*» */* «*Title*» */* «*Editor*» */* «*Publisher*» в том же направлении, исчезает и передаёт управление первой экранной кнопке. Отображение введённой информации демонстрировать при выборе из экранного списка <*select*><*/select*> определённой позиции, одного из двух элементов «показать» / «скрыть», оба из которых показаны на экране (обычный список, не выпадающий).

****

Рисунок 1 – Ограничения в жёлтых клетках таблиц

# **3. Технология локального (*LSPWA*) подключения фреймворка *Vue.js.***

# 

Рисунок 2 – Технология локального (*LSPWA*) подключения фреймворка *Vue.js.*

**4. Таблица соответствия переменных и методов, используемых в *web*-приложениях.**

В ходе выполнения работы были использованы «говорящие» переменные.

# **5. Параметрическая подстановка.**

# **5.1 Составление сети Петри технологического процесса внутри веб-страницы приложения 1.**

# 

Рисунок 3 – Сеть Петри технологического процесса веб-страницы приложения 1

**Описание сети Петри**

- состояния (*states*)

s1 – форма ожидает действий пользователя

- действия (*effects*)

e1 – появление подсказки «Сайт-визитка Черкас Н.В. и Родин К.П»

e2 – появление подсказки «Сайт-визитка Черкас Н.В. и Родин К.П»

e3 – открытие веб-страницы сайта-визитки в новой вкладке браузера

e4 – открытие веб-страницы сайта-визитки в новой вкладке браузера

e5 – закрытие формы

- события (*prompts*)

p1 – курсор наведён на гиперссылку «Перейти к сайту-визитке»

p2 – курсор наведён на гиперссылку «Открыть портфолио»

p3 – нажата гиперссылка «Перейти к сайту-визитке»

p4 – нажата гиперссылка «Открыть портфолио»

p5 – нажата кнопка закрытия веб-страницы

# **5.2 Листинг веб-страницы приложения 1.**

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

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

<title>Параметрические подстановки</title>

<script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>

<style>

body { font-family: Arial, sans-serif; padding: 20px; }

.app { max-width: 600px; margin: 0 auto; }

.link-container { margin: 15px 0; padding: 10px; border: 1px solid #ddd; }

.tooltip { position: relative; display: inline-block; }

.tooltip-text {

visibility: hidden;

background-color: #333;

color: white;

padding: 5px;

border-radius: 4px;

position: absolute;

bottom: 125%;

left: 50%;

transform: translateX(-50%);

white-space: nowrap;

z-index: 10;

}

.tooltip:hover .tooltip-text { visibility: visible; }

</style>

</head>

<body>

<div id="app1" class="app">

<h2>Приложение 1: Параметрические подстановки</h2>

<div class="link-container">

<h3>Ссылка 1:</h3>

<tooltip-link

:href="linkData.url"

:text="linkData.text1"

:target="linkData.target"

:tooltip="linkData.tooltip">

</tooltip-link>

</div>

<div class="link-container">

<h3>Ссылка 2:</h3>

<tooltip-link

:href="linkData.url"

:text="linkData.text2"

:target="linkData.target"

:tooltip="linkData.tooltip">

</tooltip-link>

</div>

<p>Обе ссылки ведут на один ресурс: <strong>{{ linkData.tooltip }}</strong></p>

</div>

<script>

const { createApp } = Vue;

const app1 = createApp({

data() {

return {

linkData: {

url: '13\_CherkasNV\_RodinKP.html', // единый ресурс

text1: 'Перейти к сайту-визитке',

text2: 'Открыть портфолио',

target: '\_blank',

tooltip: 'Сайт-визитка Черкас Н.В. и Родин К.П.'

}

}

}

});

app1.component('tooltip-link', {

props: ['href', 'text', 'target', 'tooltip'],

template: `

<div class="tooltip">

<a :href="href" :target="target">{{ text }}</a>

<span class="tooltip-text">{{ tooltip }}</span>

</div>

` });

app1.mount('#app1');

</script>

</body>

</html>

# **5.3 Результат отображения веб-страницы приложения 1 в браузере.**

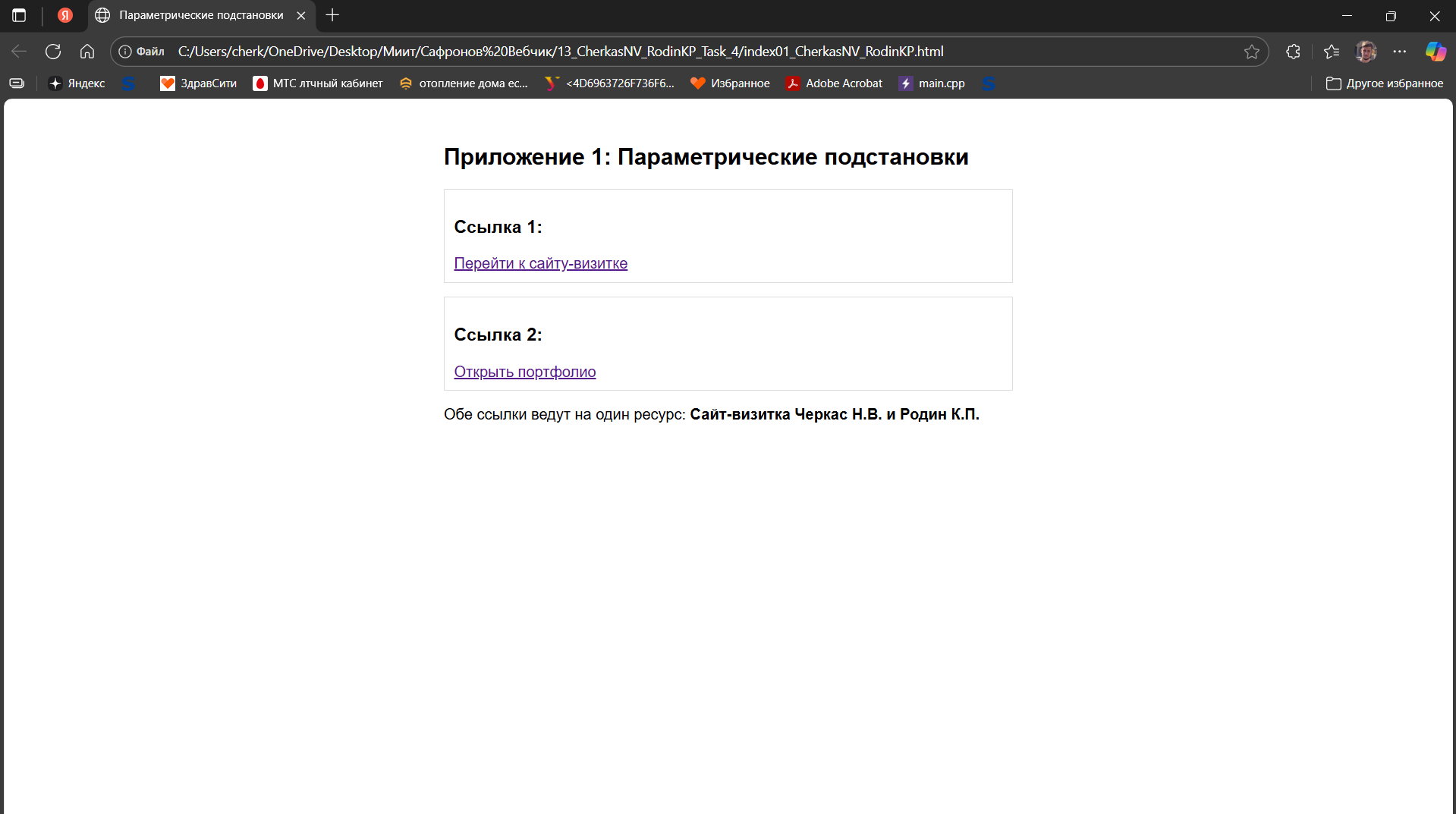


Рисунок 4 – Результат отображения в браузере веб-страницы приложения 1

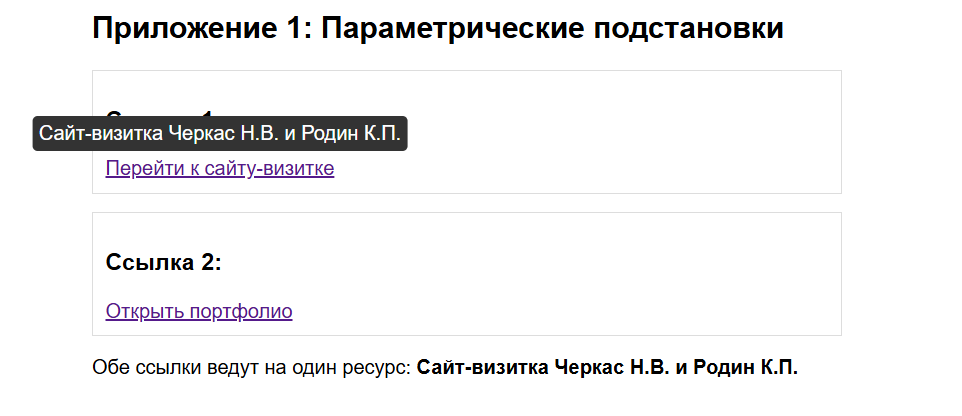


Рисунок 5 – Появление подсказки при наведении курсора на гиперссылку «Перейти к сайту-визитке»

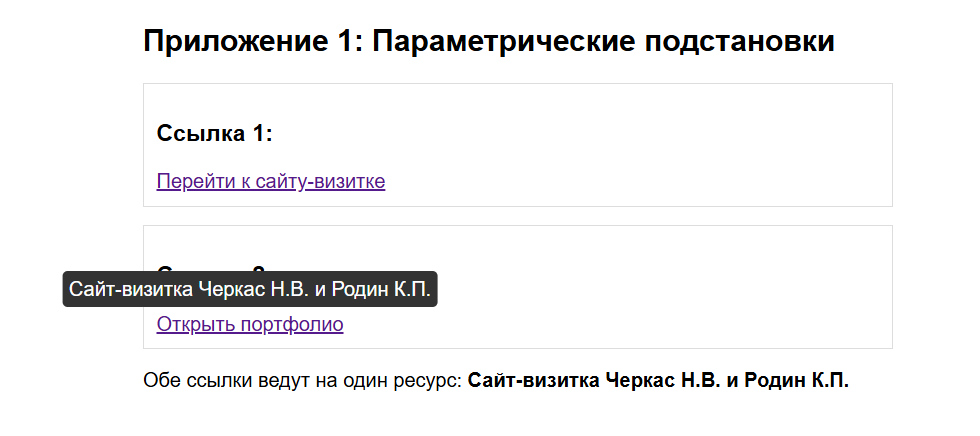


Рисунок 6 – Появление подсказки при наведении курсора на гиперссылку «Открыть портфолио»

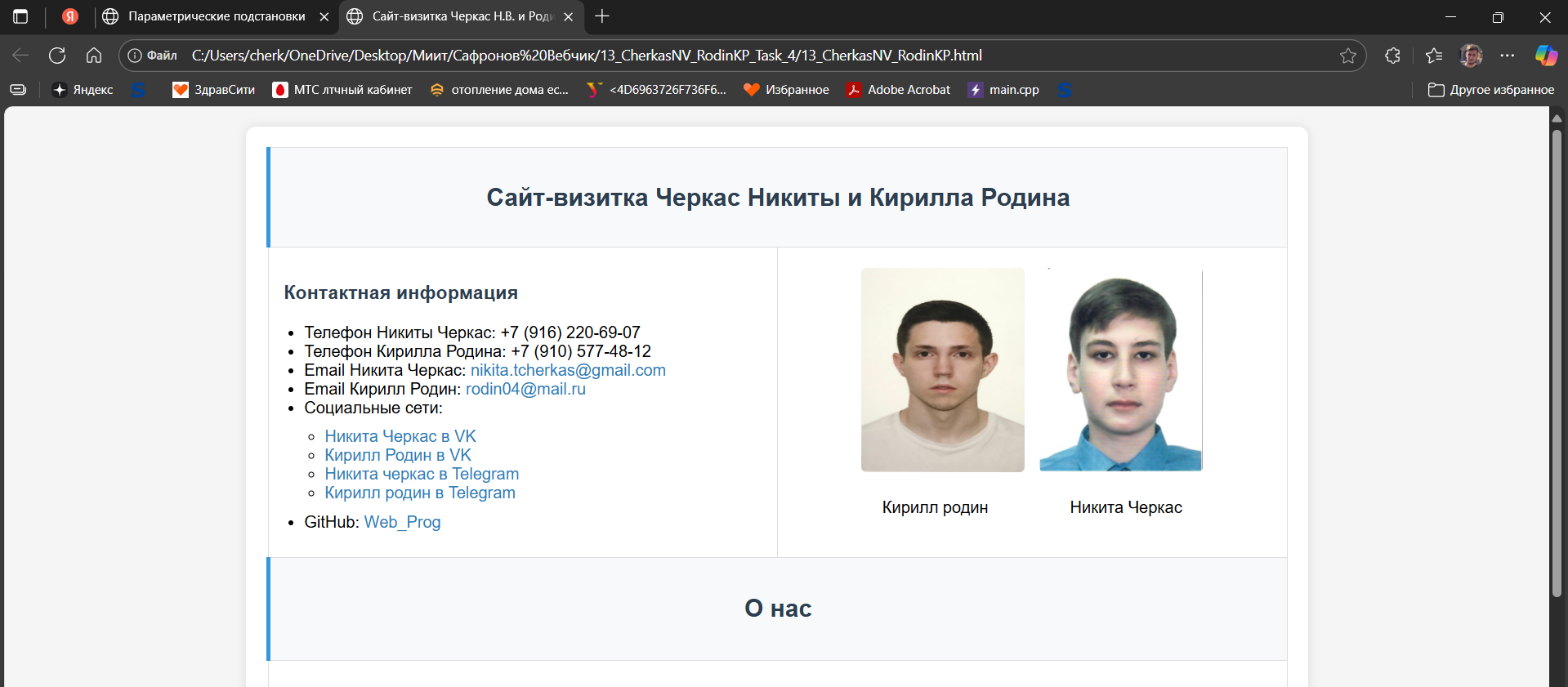


Рисунок 7 – Открытие сайта-визитки при нажатии на гиперссылку «Перейти к сайту-визитке»

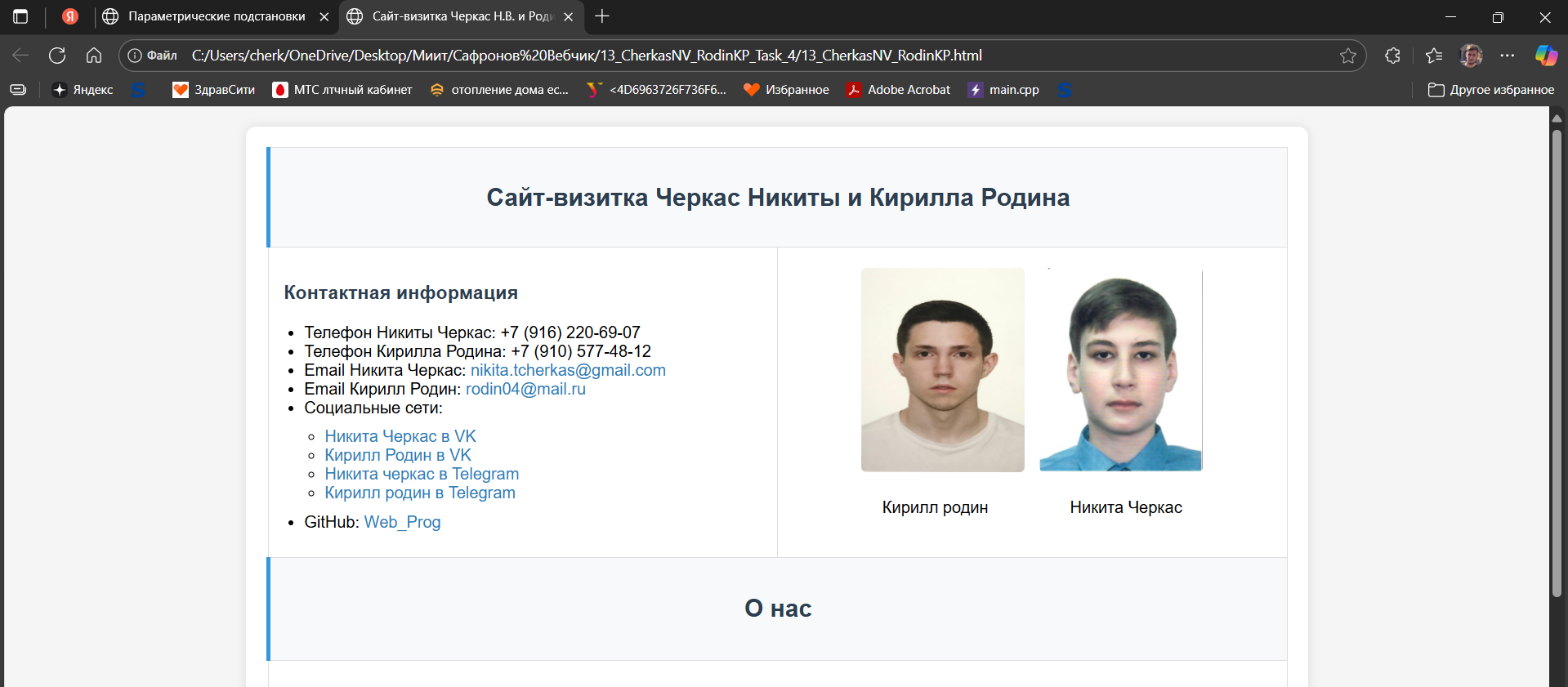


Рисунок 8 – Открытие сайта-визитки при нажатии на гиперссылку «Открыть портфолио»

# **6. Методы и события.**

# **6.1 Составление сети Петри технологического процесса внутри веб-страницы приложения 2.**

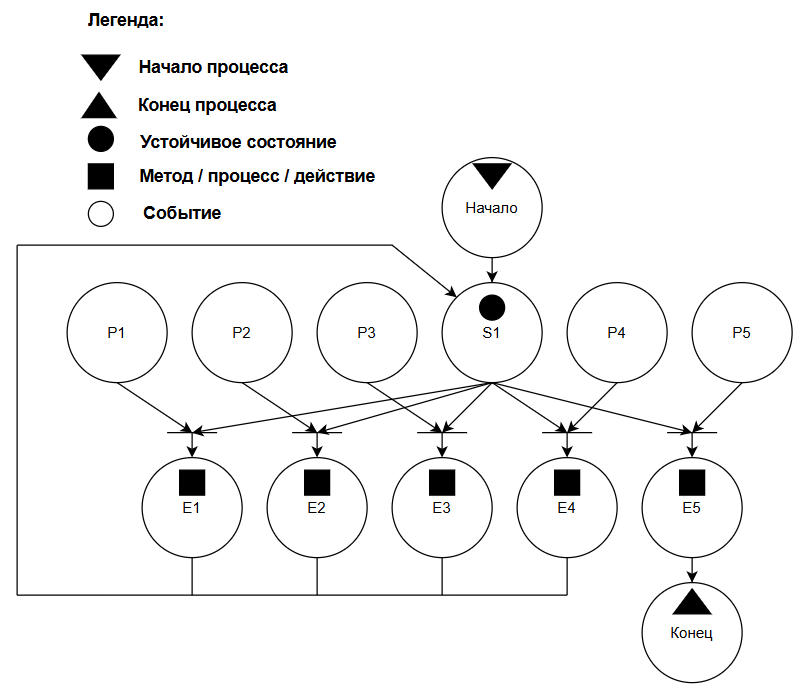
****

Рисунок 9 – Сеть Петри технологического процесса веб-страницы приложения 2

**Описание сети Петри**

- состояния (*states*)

s1 – форма ожидает действий пользователя

- действия (*effects*)

e1 – смена вида нумератора для элементов списка на предыдущий

e2 – смена вида нумератора для элемента списка на следующий

e3 – уменьшение числового значения начала списка

e4 – увеличение числового значения начала списка

e5 – закрытие формы

- события (*prompts*)

p1 – нажата кнопка «Назад»

p2 – нажата кнопка «Вперёд»

p3 – нажата кнопка «Меньше»

p4 – нажата кнопка «Больше»

p5 – нажата кнопка закрытия веб-страницы

**6.2 Листинг веб-страницы приложения 2.**

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

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

<title>Методы и события</title>

<script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>

<style>

body { font-family: Arial, sans-serif; padding: 20px; background: #f5f5f5; }

.app { max-width: 600px; margin: 20px auto; background: white; padding: 20px; border-radius: 10px; box-shadow: 0 0 10px rgba(0,0,0,0.1); }

.controls { margin: 20px 0; display: flex; gap: 10px; flex-wrap: wrap; }

button { padding: 10px 15px; border: 1px solid #ddd; background: #f8f9fa; cursor: pointer; border-radius: 5px; }

button:hover { background: #e9ecef; }

button:disabled { background: #6c757d; color: white; cursor: not-allowed; }

.list-container { margin: 20px 0; padding: 15px; border: 1px solid #ddd; border-radius: 5px; }

li { margin: 5px 0; padding: 5px; }

</style>

</head>

<body>

<div id="app2" class="app">

<h2>Приложение 2: Методы и события</h2>

<div class="controls">

<button v-on:click="prevType" v-bind:disabled="currentTypeIndex === 0">Назад</button>

<button v-on:click="nextType" v-bind:disabled="currentTypeIndex === markerTypes.length - 1">Вперёд</button>

<button v-on:click="decreaseStart" v-bind:disabled="startValue <= 0">Меньше</button>

<button v-on:click="increaseStart">Больше</button>

</div>

<div class="list-container">

<h4>Текущие настройки:</h4>

<p><strong>Тип нумерации:</strong> {{ currentMarkerType }}</p>

<p><strong>Начальное значение:</strong> {{ startValue }}</p>

<h4>Список:</h4>

<ol v-bind:start="startValue" v-bind:style="{ listStyleType: currentMarkerType }">

<li>Первый элемент списка</li>

<li>Второй элемент списка</li>

<li>Третий элемент списка</li>

</ol>

</div>

</div>

<script>

const { createApp } = Vue;

const app2 = createApp({

data() {

return {

markerTypes: ['decimal', 'upper-roman', 'lower-alpha', 'upper-alpha'],

currentTypeIndex: 0,

startValue: 1

}

},

computed: {

currentMarkerType() {

return this.markerTypes[this.currentTypeIndex];

}

},

methods: {

nextType() {

if (this.currentTypeIndex < this.markerTypes.length - 1) {

this.currentTypeIndex++;

}

},

prevType() {

if (this.currentTypeIndex > 0) {

this.currentTypeIndex--;

}

},

increaseStart() {

this.startValue++;

},

decreaseStart() {

if (this.startValue > 0) {

this.startValue--;

}

}

}

});

app2.mount('#app2');

</script>

</body>

</html>

**6.3 Результат отображения веб-страницы приложения 2.**

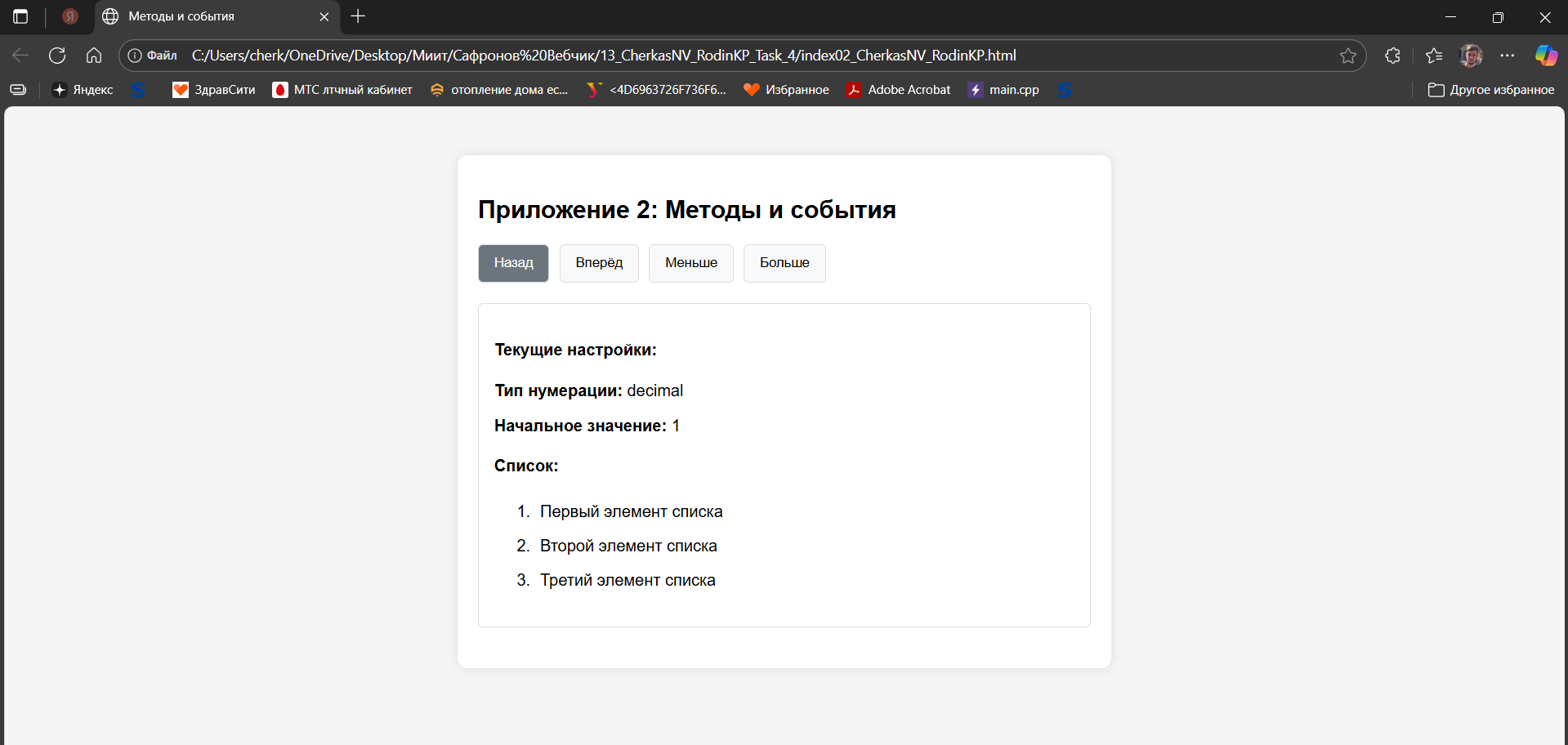


Рисунок 10 – Результат отображения в браузере веб-страницы приложения 2

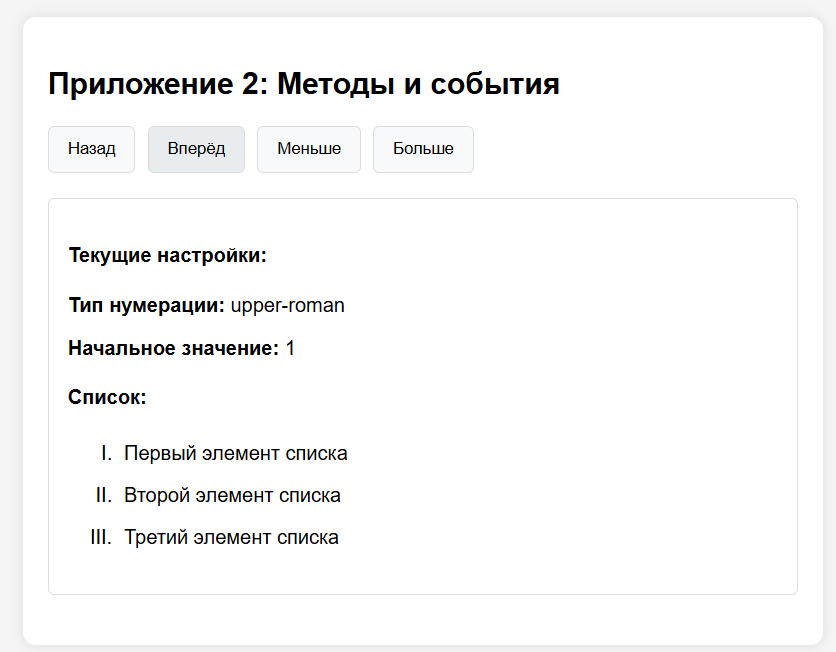


Рисунок 11 – Результат начатия кнопки «Впрёд»

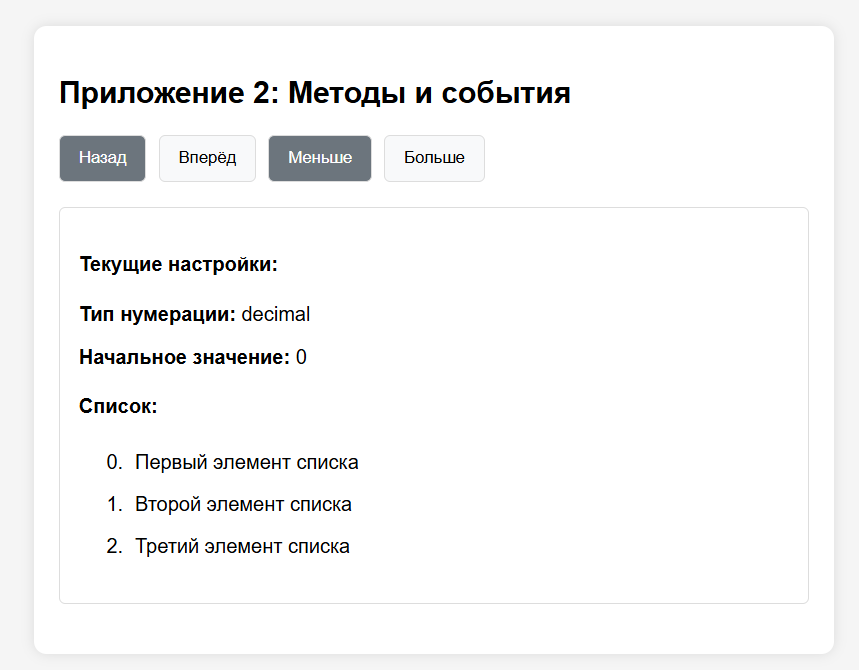


Рисунок 12 – Результат нажатия кнопки «Меньше» и кнопки «Назад»

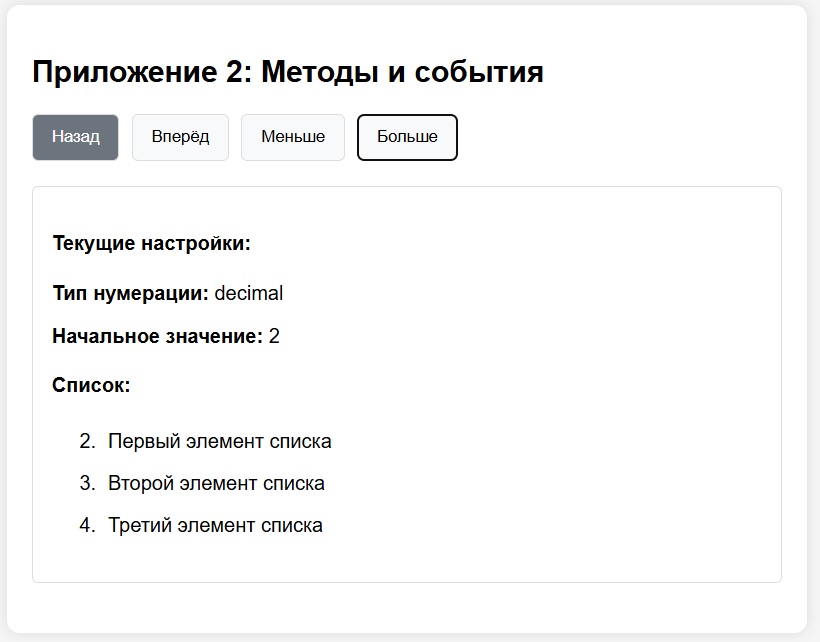


Рисунок 13 – Результат нажатия кнопки «Больше»

# **7. Параметрическая связка.**

# **7.1 Составление сети Петри технологического процесса внутри веб-страницы приложения 3.**

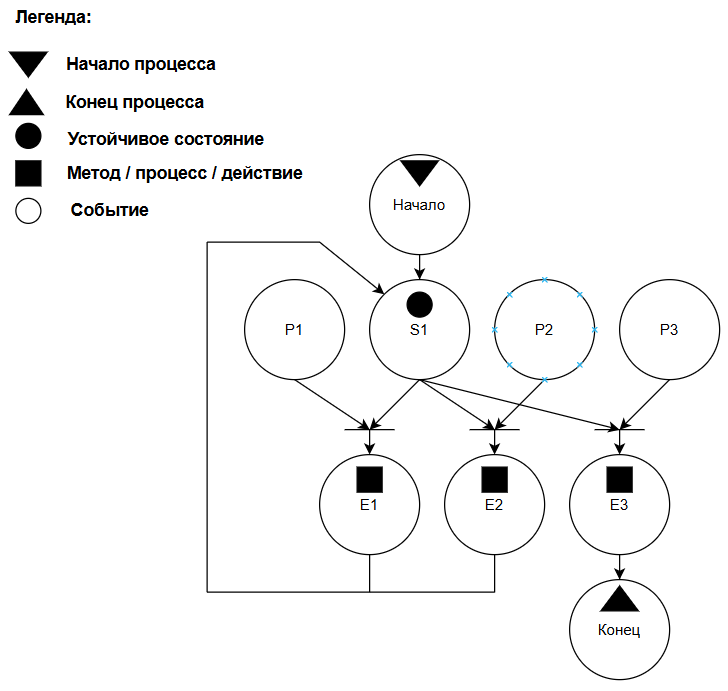


Рисунок 14 – Сеть Петри технологического процесса веб-страницы приложения 3

**Описание сети Петри**

- состояния (*states*)

s1 – форма ожидает действий пользователя

- действия (*effects*)

e1 – выделение следующей (нижней) позиции в списке

e2 – выделение предыдущей (верхней) позиции в списке

e3 – закрытие формы

- события (*prompts*)

p1 – нажата кнопка «Предыдущий»

p2 – нажата кнопка «Следующий»

p3 – нажата кнопка закрытия веб-страницы

# **7.2 Листинг веб-страницы приложения 3.**

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

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

<title>Параметрическая связка</title>

<script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>

</head>

<body style="font-family: Arial, sans-serif; padding: 20px; background: #f5f5f5;">

<div id="app3"

style="max-width: 600px; margin: 20px auto; background: white; padding: 20px; border-radius: 10px; box-shadow: 0 0 10px rgba(0,0,0,0.1);">

<h2>Приложение 3: Параметрическая связка</h2>

<div style="margin: 20px 0; display: flex; gap: 10px;">

<button

@click="selectPrev"

:disabled="selectedIndex === 0"

style="padding: 10px 15px; border: 1px solid #ddd; background: #f8f9fa; cursor: pointer; border-radius: 5px;"

:style="selectedIndex === 0 ? 'background:#6c757d; color:white; cursor:not-allowed;' : 'background:#f8f9fa;'"

>Предыдущий</button>

<button

@click="selectNext"

:disabled="selectedIndex === 4"

style="padding: 10px 15px; border: 1px solid #ddd; background: #f8f9fa; cursor: pointer; border-radius: 5px;"

:style="selectedIndex === 4 ? 'background:#6c757d; color:white; cursor:not-allowed;' : 'background:#f8f9fa;'"

>Следующий</button>

</div>

<div style="margin: 10px 0; padding: 10px; background: #e9ecef; border-radius: 5px;">

<strong>Текущая позиция:</strong> {{ selectedIndex + 1 }} из 5

</div>

<div style="margin: 20px 0;">

<select v-model="selectedIndex" size="5" style="width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 5px; background: #f8f9fa; ; pointer-events: none;">

<option :value="0" :style="selectedIndex === 0 ? 'background:#007bff; color:white;' : ''">Первый элемент списка</option>

<option :value="1" :style="selectedIndex === 1 ? 'background:#007bff; color:white;' : ''">Второй элемент списка</option>

<option :value="2" :style="selectedIndex === 2 ? 'background:#007bff; color:white;' : ''">Третий элемент списка</option>

<option :value="3" :style="selectedIndex === 3 ? 'background:#007bff; color:white;' : ''">Четвертый элемент списка</option>

<option :value="4" :style="selectedIndex === 4 ? 'background:#007bff; color:white;' : ''">Пятый элемент списка</option>

</select>

</div>

</div>

<script>

const { createApp } = Vue;

createApp({

data() {

return {

selectedIndex: 0

};

},

methods: {

selectNext() {

if (this.selectedIndex < 4) this.selectedIndex++;

},

selectPrev() {

if (this.selectedIndex > 0) this.selectedIndex--;

}

}

}).mount('#app3');

</script>

</body>

</html>

**7.3 Результат отображения веб-страницы приложения 3.**

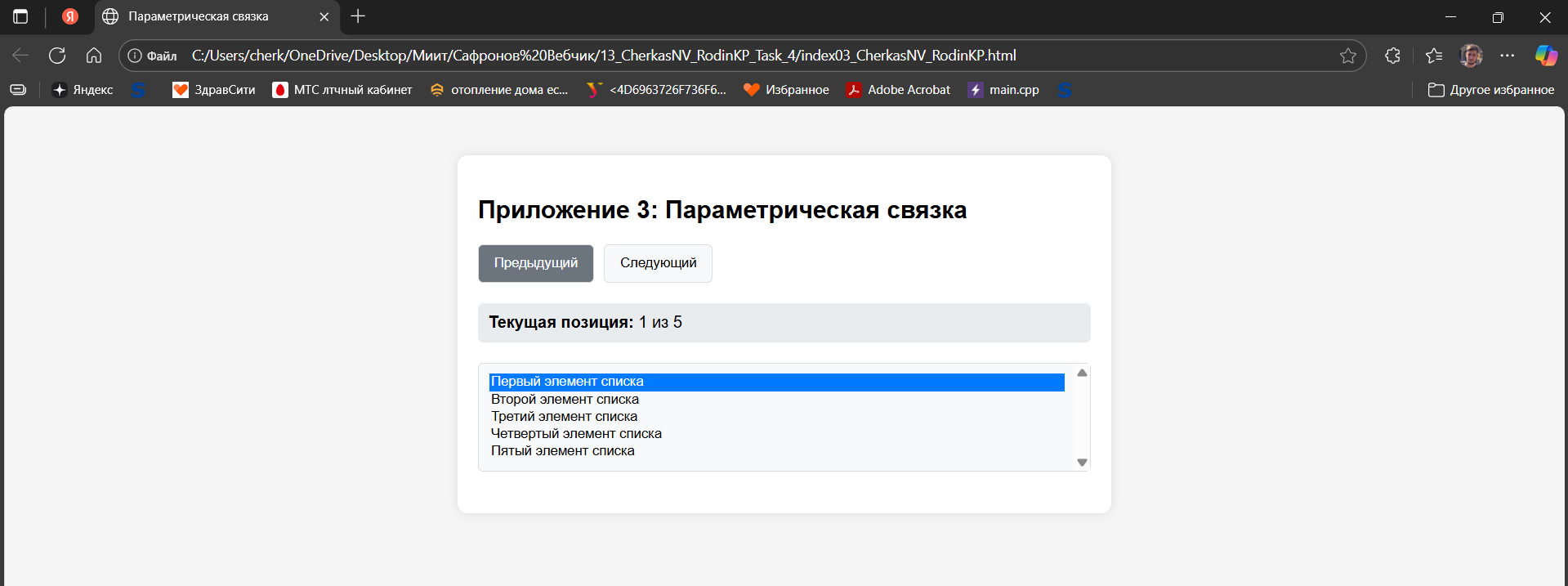


Рисунок 15 – Отображение в браузере веб-страницы приложения 3

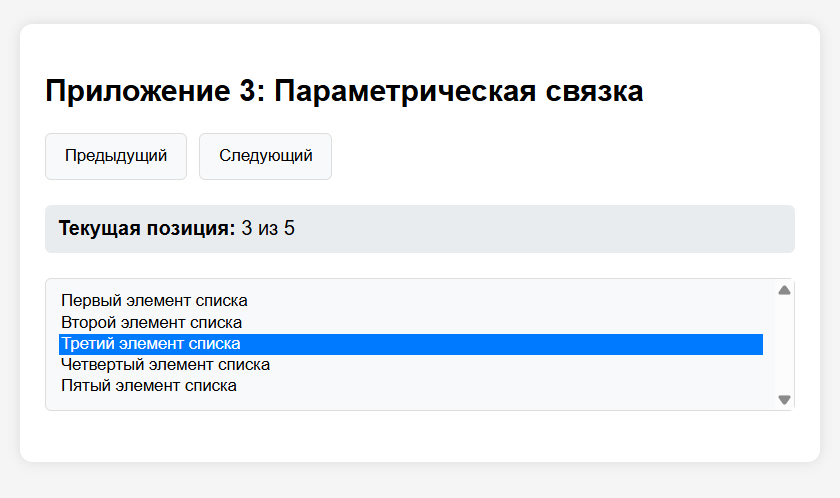


Рисунок 16 – Результат двойного нажатия кнопки «Следующий»

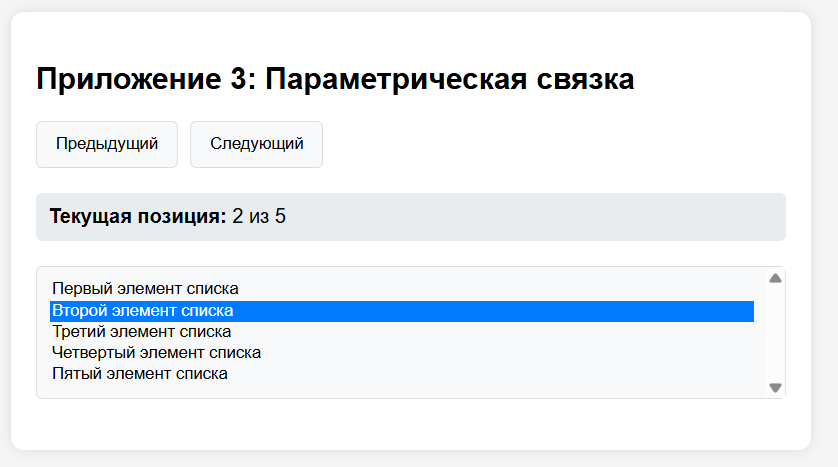


Рисунок 17 – Результат нажатия кнопки «Предыдущий»

# **8. Условная отрисовка / шаблоны.**

# **8.1 Составление сети Петри технологического процесса внутри веб-страницы приложения 4.**

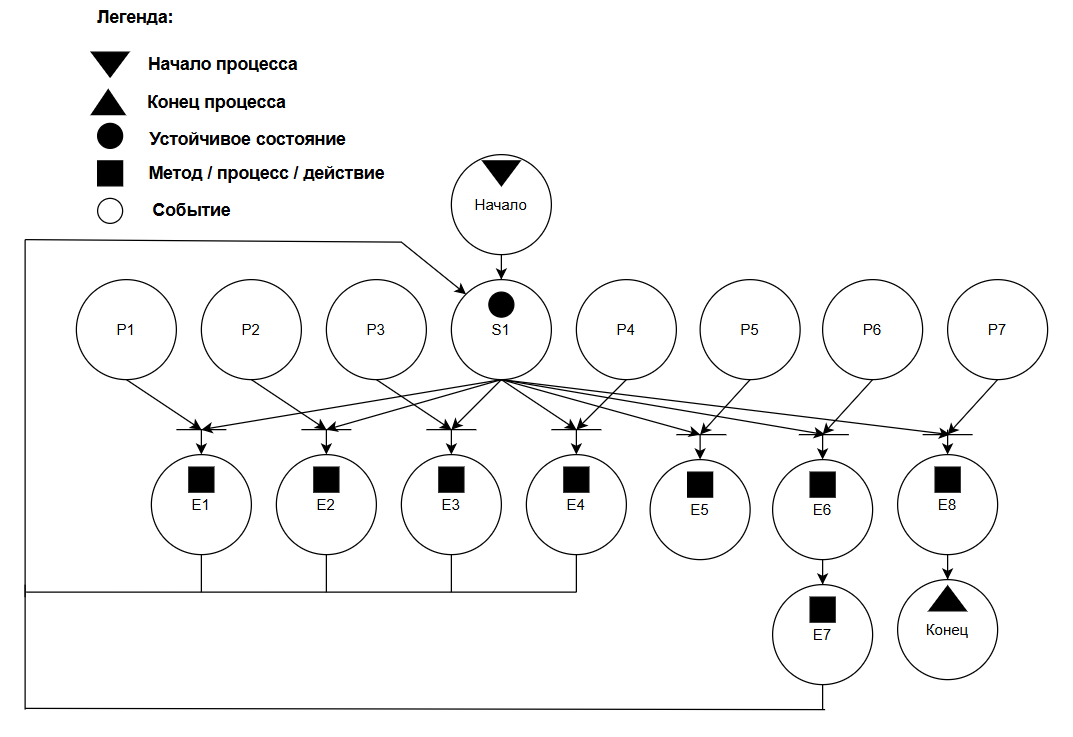


Рисунок 18 – Сеть Петри технологического процесса веб-страницы приложения 4

**Описание сети Петри**

- состояния (*states*)

s1 – форма ожидает действий пользователя

- действия (*effects*)

e1 – Появление в форме текстового курсора

e2 – Отображение введённых данных в форме

e3 – Скрытие данных, записанных в форму

e4 – Показ данных, записанных в форму

e5 – Смена шаблона

e6 – Становление кнопки «Смена шаблона» / «Смена шаблона2» невидимой

e7 – Становление кнопки «Смена шаблона2» / «Смена шаблона» видимой

e8 – закрытие формы

- события (*prompts*)

p1 – Нажатие на форму для введения данных

p2 – Ввод данных с клавиатуры

p3 – Нажатие кнопки «Скрыть»

p4 – Нажатие кнопки «Показать»

p5 – Нажатие кнопки «Смена шаблона» или «Смена шаблона2»

p6 – Нажатие кнопки «Смена шаблона» или «Смена шаблона2» четвёртый раз

p7 – нажата кнопка закрытия веб-страницы

# **8.2 Листинг веб-страницы приложения 4.**

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

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

<title>Условная отрисовка / шаблоны</title>

<script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>

</head>

<body>

<div id="app4">

<h2>Приложение 4: Условная отрисовка / шаблоны</h2>

<div style="margin: 20px 0; display: flex; gap: 10px; position: relative; height: 50px;">

<button v-if="showFirstButton" v-on:click="nextTemplate" style="position: absolute; left: 0; padding: 10px 15px;">Смена шаблона</button>

<button v-if="showSecondButton" v-on:click="nextTemplate" style="position: absolute; left: 150px; padding: 10px 15px;">Смена шаблона2</button>

</div>

<div style="margin: 20px 0; padding: 20px; border: 2px solid #007bff; border-radius: 5px; background: #f8f9fa;">

<h3>Текущий шаблон: {{ currentTemplate }}</h3>

<!-- Верхнее поле ввода -->

<div style="margin: 10px 0;">

<label style="display: block; margin-bottom: 5px; font-weight: bold;">Введите информацию:</label>

<input v-model="inputData" type="text" placeholder="Введите данные..."

style="width: 100%; padding: 8px; border: 1px solid #ddd; border-radius: 4px;">

</div>

<!-- Нижнее поле отображения -->

<div style="margin: 10px 0;">

<label style="display: block; margin-bottom: 5px; font-weight: bold;">Отображение информации:</label>

<!-- При "показать" отображаем сохранённый текст -->

<input v-if="displayMode === 'show'"

:value="savedData[currentTemplate]" readonly

style="width: 100%; padding: 8px; border: 1px solid #ddd; border-radius: 4px; background: white;">

<!-- При "скрыть" показываем пустое поле -->

<input v-if="displayMode === 'hide'"

value="" readonly

style="width: 100%; padding: 8px; border: 1px solid #ddd; border-radius: 4px; background: white;">

</div>

<!-- Список выбора режима отображения -->

<div style="margin: 15px 0;">

<label style="display: block; margin-bottom: 5px; font-weight: bold;">Режим отображения:</label>

<select v-model="displayMode" size="2" style="width: 100%; padding: 8px; border: 1px solid #ddd; border-radius: 4px; background: white;">

<option value="show">показать</option>

<option value="hide">скрыть</option>

</select>

</div>

</div>

</div>

<script>

const { createApp } = Vue;

createApp({

data() {

return {

templates: ['Author', 'Title', 'Editor', 'Publisher'],

currentTemplateIndex: 0,

inputData: '',

displayMode: 'show',

showFirstButton: true,

showSecondButton: false,

clickCount: 0,

// объект для хранения данных по каждому шаблону

savedData: {

Author: '',

Title: '',

Editor: '',

Publisher: ''

}

};

},

computed: {

currentTemplate() {

return this.templates[this.currentTemplateIndex];

}

},

watch: {

// при изменении поля ввода — сохраняем значение для текущего шаблона

inputData(newVal) {

this.savedData[this.currentTemplate] = newVal;

}

},

methods: {

nextTemplate() {

// сохраняем данные текущего шаблона

this.savedData[this.currentTemplate] = this.inputData;

// переключаем шаблон

this.currentTemplateIndex = (this.currentTemplateIndex + 1) % this.templates.length;

// обновляем поле ввода

this.inputData = this.savedData[this.currentTemplate];

// переключение кнопок каждые 4 клика

this.clickCount++;

if (this.clickCount === 4) {

this.showFirstButton = !this.showFirstButton;

this.showSecondButton = !this.showSecondButton;

this.clickCount = 0;

}

}

}

}).mount('#app4');

</script>

</body>

</html>

**8.3 Результат отображения веб-страницы приложения 4.**

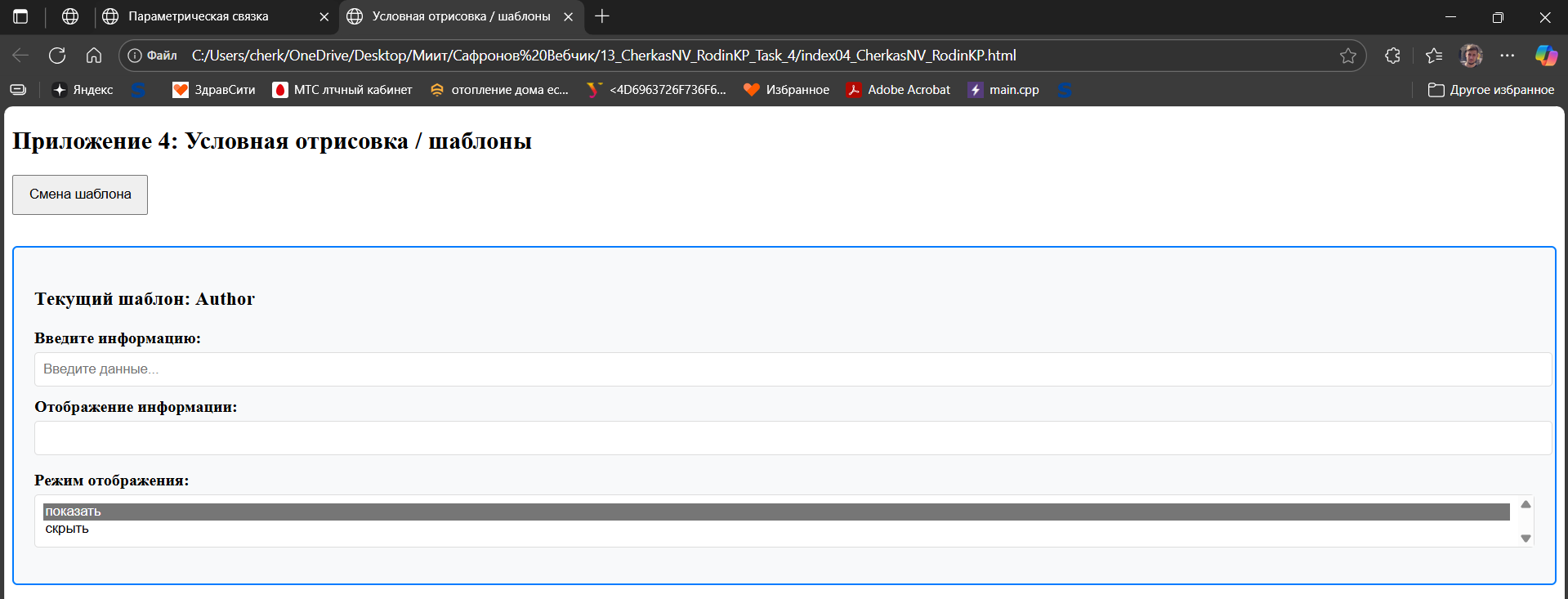
****

Рисунок 19 – Результат отображения веб-страницы приложения 4

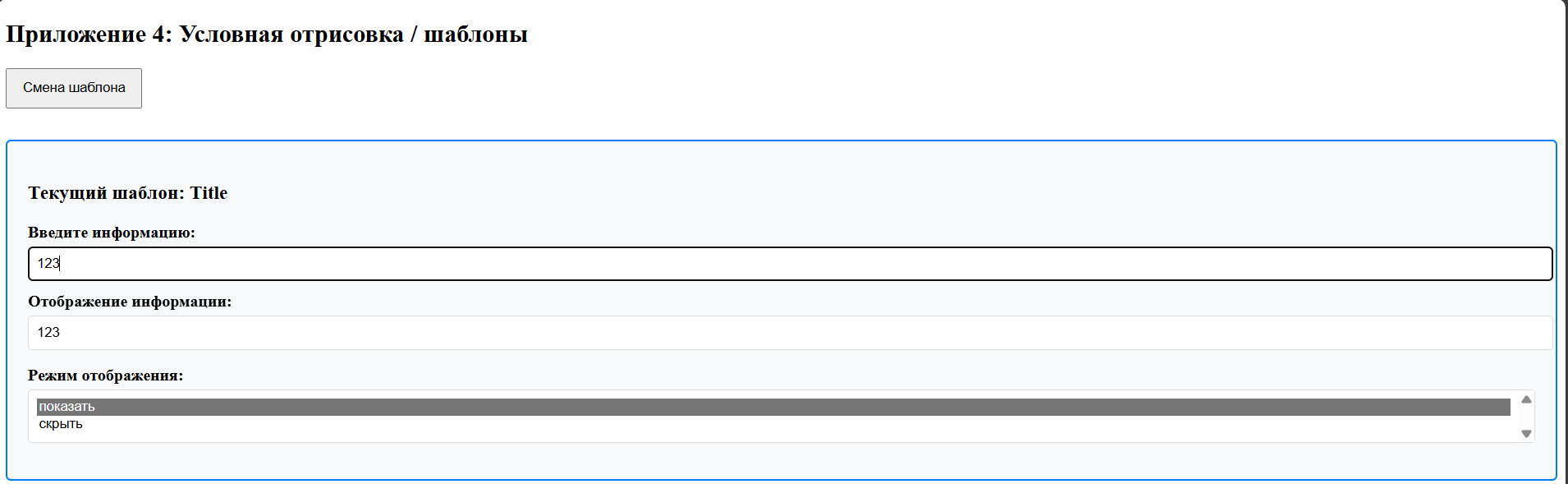


Рисунок 20 – Результат нажатия кнопки «Смена шаблона» и ввода данных в форму

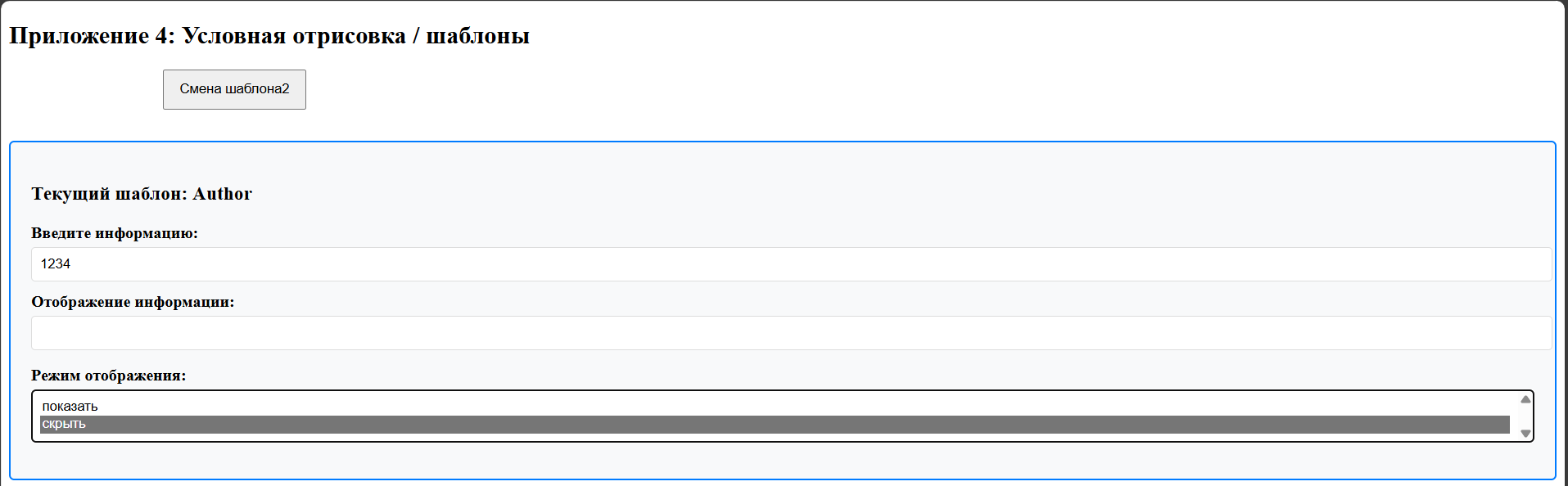


Рисунок 21 – Результат четверного нажатия кнопки «Смена шаблона» и кнопки «Скрыть»

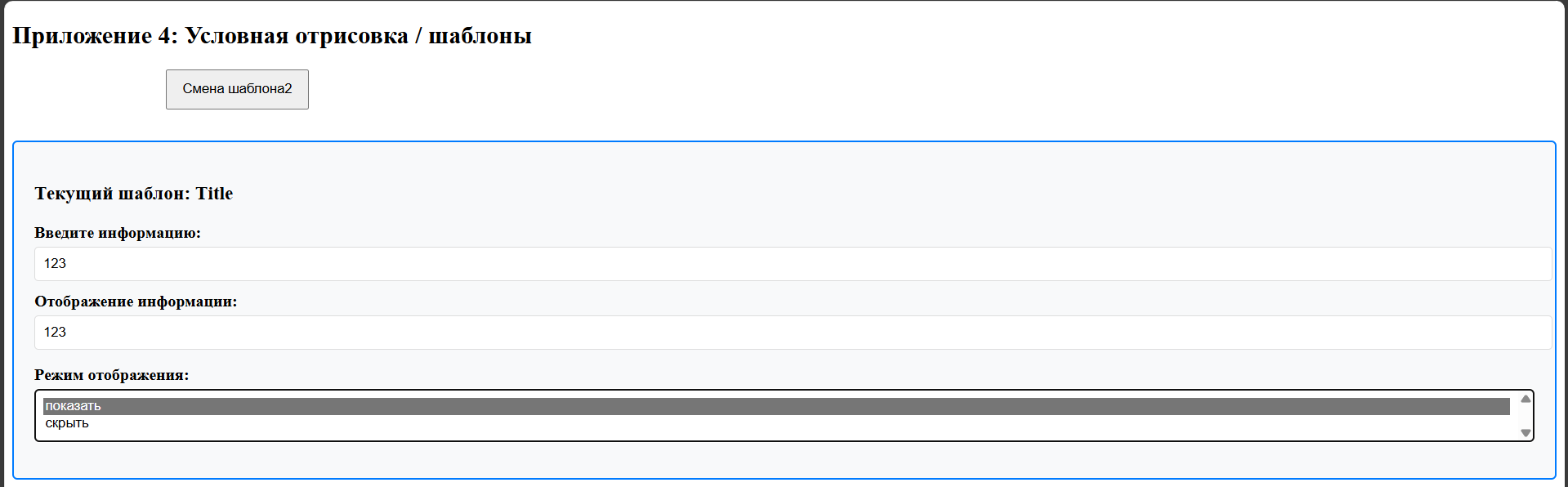


Рисунок 22 – Результат нажатия кнопки «Смена шаблона2» и кнопки «Показать»

# **9. Формулировку вывода о проделанной работе**

В рамках данной работы были получены навыки разработки простых локальных *web*-приложений (*LSPWA*) под управлением фреймворка *Vue.js* на языке *JavaScript*.