Министерство транспорта Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский университет транспорта (МИИТ)» (РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Отчёт

по практике

«Учебная практика»

1. ­Практика по получению первичных профессиональных умений и навыков, в том числе умений и навыков научно-исследовательской деятельности

Выполнил: ст. гр. ТУУ-211

Дидушок Д.А.

Проверил: доц. Сафронов А.И.

Москва­ – 2018 г.

**Задание 1.**

**Цель работы:** создать пошаговый веб-интерфейс с использованием Vue.Js

**Содержательная часть:**

**1. Основная страница:**

<html>

<head>

<meta charset="utf-8">

<title>Каталог</title>

</head>

<body>

<script src="Vue.js"></script>

<script src="comp\_sf.js"></script>

<script src="comp\_nf.js"></script>

<script src="comp\_f.js"></script>

<script src="comp\_l.js"></script>

<script type="text/x-template" id="selectfile">

<div id="myCatalog">

<input type="file" id="file">

<button @click="readFile($event)" @click="next">Далее</button>

</div>

</script>

<script type="text/x-template" id="noFormat">

<div id="myCatalog">

<b>Вывод без форматирования</b>

<br>

<li v-for="el in this.$root.fileData">{{ el }}</li>

<br>

<button @click="back">Назад</button>

<button @click="next">Далее</button>

</div>

</script>

<script type="text/x-template" id="Format">

<div id="myCatalog">

<b>Вывод с форматированием: </b>

<li v-for="el in this.$root.fileData">{{el.id}}. <b>Название товара:</b> <i>{{el.productname}}</i> <b> Пол:</b> <i>{{el.sex}}</i> <b>Описание:</b> <i>{{ el.description.effect}}</i> <b> Цена: </b> <i>{{el.price}}</i> </li>

<br>

<button @click="back">Назад</button>

<button @click="next">Далее</button>

</div>

</script>

<script type="text/x-template" id="Spisok">

<div id="myCatalog">

<b>Выберите элемент из списка</b>

<br>

<select v-model="selected">

<option v-for="el in this.$root.fileData" v-bind:value="el.description.effect"> {{el.productname}} </option>

</select>

<span v-if="selected!==''"> <b>Описание:</b> <i>{{selected}}</i> </span>

<br>

<button @click="back">Назад</button>

</div>

</script>

<div id="myCatalog">

<component v-bind:is="getPage"></component>

</div>

<script>

myVueCatalog = new Vue(

{

el: '#myCatalog',

data:

{

fileData: '',

pageN: 'fSelect'

},

computed:

{

getPage: function()

{

if (this.pageN == 'fSelect')

{

return 'selectf';

}

else

{

if (this.pageN == 'NForm')

{

return 'noform';

}

else

{

if (this.pageN == 'Form')

{

return 'format';

}

else

{

if (this.pageN == 'Spisok')

{

return 'nspis';

}

}

}

}

}

},

methods:

{

},

})

</script>

</body>

</html>

2. Сценарий страницы с форматированным списком:

Vue.component('format',

{

template: '#Format',

methods:

{

next: function()

{

myVueCatalog.pageN = 'Spisok';

},

back: function()

{

myVueCatalog.pageN = 'NForm';

},

}

})

3. Сценарий страницы с обычным списком:

Vue.component('nspis',

{

template: '#Spisok',

props:

{

data: '',

},

data: function()

{

return {

selected: '',

}

},

methods:

{

back: function()

{

myVueCatalog.pageN = 'Form';

},

}

})

4. Сценарий страницы с неформатированным списком:

Vue.component('noform',

{

template: '#noFormat',

methods:

{

next: function()

{

myVueCatalog.pageN = 'Form';

},

back: function()

{

myVueCatalog.pageN = 'fSelect';

},

}

})

5. Сценарий страницы выбора файла:

Vue.component('selectf',

{

template: '#selectfile',

methods:

{

readFile: function(e)

{

var vm = this;

var file = e.target.ownerDocument.getElementById('file').files[0];

var reader = new FileReader();

reader.onload = function()

{

vm.$root.fileData = JSON.parse(reader.result);

}

reader.readAsText(file);

},

next: function()

{

myVueCatalog.pageN = 'NForm';

},

}

})

**Задание 2.**

**Цель работы:** создать «умный калькулятор» с использованием Vue.Js

**Содержательная часть:**

1. Основная страница:

<HTML>

<TITLE>Проверка комбинированного списка</TITLE>

<BODY>

<SCRIPT SRC="Vue.js"></SCRIPT>

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

<div class="calc-body">

<div class="calc-body-screen">

<div class="a"><span id="answertile">{{ans}}</span></div>

</div>

<div class="calc-body-btn" @click="mc()"><span>MC</span></div>

<div class="calc-body-btn" @click="mr()"><span>MR</span></div>

<div class="calc-body-btn" @click="ms()"><span>MS</span></div>

<div class="calc-body-btn" @click="m\_plus()"><span>M+</span></div>

<div class="calc-body-btn" @click="m\_minus()"><span>M-</span></div>

<div id="btn\_back" class="calc-body-btn" @click="back()"><span><</span></div>

<div id="btn\_clear\_all" class="calc-body-btn" @click="clear()"><span>CE</span></div>

<div id="btn\_clear" class="calc-body-btn" @click="clear\_all()"><span>C</span></div>

<div id="btn\_rvnag" class="calc-body-btn" @click="rvnag()"><span>±</span></div>

<div id="btn\_sqrt" class="calc-body-btn" @click="sqrt()"><span>v</span></div>

<div class="calc-body-btn btnnum" @click="getValue(7)"><span>7</span></div>

<div class="calc-body-btn btnnum" @click="getValue(8)"><span>8</span></div>

<div class="calc-body-btn btnnum" @click="getValue(9)"><span>9</span></div>

<div class="calc-body-btn btnnum" @click="getValue('/')"><span>/</span></div>

<div class="calc-body-btn btnnum" @click="proc()"><span>%</span></div>

<div class="calc-body-btn btnnum" @click="getValue(4)"><span>4</span></div>

<div class="calc-body-btn btnnum" @click="getValue(5)"><span>5</span></div>

<div class="calc-body-btn btnnum" @click="getValue(6)"><span>6</span></div>

<div class="calc-body-btn btnnum" @click="getValue('\*')"><span>\*</span></div>

<div id="btn\_reverse" class="calc-body-btn" @click="reverse()"><span>1/x</span></div>

<div class="left">

<div class="calc-body-btn btnnum" @click="getValue(1)"><span>1</span></div>

<div class="calc-body-btn btnnum" @click="getValue(2)"><span>2</span></div>

<div class="calc-body-btn btnnum" @click="getValue(3)"><span>3</span></div>

<div class="calc-body-btn btnnum" @click="getValue('-')"><span>-</span></div>

<div class="calc-body-btn zero btnnum" @click="getValue(0)"><span>0</span></div>

<div class="calc-body-btn btnnum" @click="getValue('.')"><span>.</span></div>

<div class="calc-body-btn btnnum" @click="getValue('+')"><span>+</span></div>

</div>

<div id="btn\_cal\_all" class="calc-body-btn qu" @click="calc\_all()"><span>=</span></div>

</div>

</div>

<SCRIPT>

var global\_cal = {};

global\_cal.ans = '';

global\_cal.memory = '';

var vm = new Vue(

{

el: "#app",

data: global\_cal,

methods:

{

getValue:function(item)

{

var countElAns = global\_cal.ans.length;

if (global\_cal.ans == 'Ошибка')

{

global\_cal.ans = '';

}

if (countElAns < 11)

{

global\_cal.ans = global\_cal.ans + item

}

else

{

global\_cal.ans = 'Ошибка';

}

},

mc:function()

{

global\_cal.memory = '';

},

m\_minus:function()

{

global\_cal.memory = ('' + (eval(global\_cal.memory) - eval(global\_cal.ans))).substr(0,9);

},

m\_plus:function()

{

global\_cal.memory = ('' + (eval(global\_cal.memory) + eval(global\_cal.ans))).substr(0,9);

},

ms:function()

{

global\_cal.memory = global\_cal.ans;

},

mr:function()

{

global\_cal.ans = global\_cal.memory;

},

rvnag:function()

{

global\_cal.ans = ('' + (parseFloat(eval(global\_cal.ans)) \* -1)).substr(0,9);

},

sqrt:function()

{

if (eval(global\_cal.ans) >= 0)

{

global\_cal.ans = ('' + (Math.sqrt(parseFloat(eval(global\_cal.ans))))).substr(0,9);

}

else

{

global\_cal.ans = 'Ошибка';

}

},

proc:function()

{

var zn = '';

if (global\_cal.ans.indexOf('+') != -1)

{

zn = '+';

}

else if (global\_cal.ans.indexOf('-') != -1)

{

zn = '-';

}

else if (global\_cal.ans.indexOf('\*') != -1)

{

zn = '\*';

}

else

{

zn = '/'

}

var temp = global\_cal.ans.substr(global\_cal.ans.indexOf(zn) + 1, global\_cal.ans.length);

temp = ('' + (parseFloat(global\_cal.ans.substr(0, global\_cal.ans.indexOf(zn))) \* parseFloat(temp)) / 100).substr(0,7);

global\_cal.ans = global\_cal.ans.substr(0, global\_cal.ans.indexOf(zn) + 1) + temp;

},

reverse:function()

{

global\_cal.ans = ('' + (1/parseFloat(eval(global\_cal.ans)))).substr(0,9);

},

back:function()

{

global\_cal.ans = global\_cal.ans.substr(0, global\_cal.ans.length - 1);

},

clear:function()

{

global\_cal.ans = '';

},

clear\_all:function()

{

global\_cal.memory = '';

global\_cal.ans = '';

},

calc\_all:function()

{

global\_cal.ans = ('' + eval(global\_cal.ans)).substr(0,9);

},

}

});

</SCRIPT>

<STYLE TYPE="text/css">

html,body

{

width: 100%;

height: 100%;

padding: 0;

margin: 0;

}

.calculator

{

width: 100%;

height: 100%;

min-height: 600px;

background-image: linear-gradient(to bottom, #038FA1, #023E6E);

display: flex;

justify-content: center;

align-items: center;

}

.calc-body

{

width: 85%;

max-width: 250px;

height: 375px;

background-image: linear-gradient(to bottom, #004D59, #012024);

padding: 10px;

border-radius: 10px;

box-shadow: 0 25px 35px -15px rgba(3, 28, 31, .85);

/\*flex\*/

display: flex;

flex-wrap: wrap;

justify-content: space-between;

align-items: stretch;

align-content: space-between;

}

.calc-body-screen

{

flex: 1 0 100%;

height: 24%;

color: #A8EEFF;

box-shadow: inset 0 0 0 1px #225861;

display: flex;

flex-direction: column;

justify-content: space-between;

text-align: right;

}

.calc-body-screen .a

{

line-height: 80px;

width: 100%;

padding: 0 5%;

box-sizing: border-box;

font-size: 38px;

}

.calc-body-btn

{

cursor: pointer;

opacity: .6;

font-size: 12px;

flex: 0 1 18%;

height: 11%;

color: #A8EEFF;

box-shadow: inset 0 0 0 1px #225861;

text-align: center;

border-radius: 5px;

display: flex;

justify-content: center;

align-items: center;

}

.calc-body-btn:hover

{

background-color: #225861;

}

.calc-body-btn.qu

{

height: 23%;

}

.left

{

display: flex;

flex: 0 0 79.5%;

height: 23%;

align-content: space-between;

justify-content: space-between;

flex-wrap: wrap;

}

.left .calc-body-btn

{

height: 46%;

flex: 0 0 23%;

}

.left .calc-body-btn.zero

{

flex: 0 1 48.5%;

}

</STYLE>

</BODY>

</HTML>

**Вывод:** изучены основы фреймворка Vue.js. Создано и отлажено работоспособное программное обеспечение под веб.