2 Лабораторная работа №2
Исследование возможностей программирования на стороне клиента. Основы языка JavaScript

2.1. Цель работы

Исследовать особенности написания программ для приложений на стороне клиента. Изучить основы языка JavaScript и объектной модели браузера. Приобрести практические навыки проверки HTML-форм с использованием JavaScript.

2.2. Постановка задачи

1. Модифицировать страницу «Фотоальбом» (использовать HTML-страницы, разработанные при выполнении предыдущей лабораторной работы), реализовав вывод таблицы, содержащей фото, с использованием операторов циклов. Значения имен файлов фото и подписей к фото предварительно разместить в массивах fotos и titles.
2. Модифицировать страницу «Мои интересы», реализовав вывод списков с использованием JavaScript-функции с переменным числом аргументов.
3. Добавить на страницах «Контакт» и «Тест по дисциплине «...»» функции проверки заполненности форм. В случае если какое-либо из полей формы осталось незаполненным при нажатии на кнопку отправить, вывести сообщение об ошибке и установить фокус на незаполненный элемент.
4. Добавить на странице «Контакт» текстовое поле «Телефон». Для полей «Фамилия Имя Отчество» и «Телефон» добавить функции специфической проверки значений. В случае если какое-либо из полей формы заполнено не верно, при нажатии на кнопку отправить, вывести сообщение об ошибке и установить фокус на неверно заполненный элемент. Формат правильных значений полей: Фамилия Имя Отчество – введено три слова, разделенные

одним пробелом; Телефон – строка может состоять только из цифр; начинаться только с последовательности «+7» или «+3»; не содержит пробелов; количество цифр в строке от 9 до 11.

5. Добавить на странице «Тест по дисциплине «...»» функции специфической проверки значений полей в соответствии с вариантом задания, представленном в таблице 2.4. В случае не выполнения условия сформировать сообщение об ошибке и установить фокус на неверно заполненный элемент ввода.

2.3. Ход выполнения работы

Весь написанный в процессе выполнения лабораторной работы код скрипта на JavaScript представлен в приложении Б. Некоторые страницы сайта, созданные в лабораторной работе №1, были изменены для выполнения поставленных задач текущей работы.

В страницу «Фотоальбом» был внедрён скрипт на JavaScript, автоматически заполняющий таблицу после загрузки сайта. Рисунок 1 демонстрирует корректность составления таблицы.

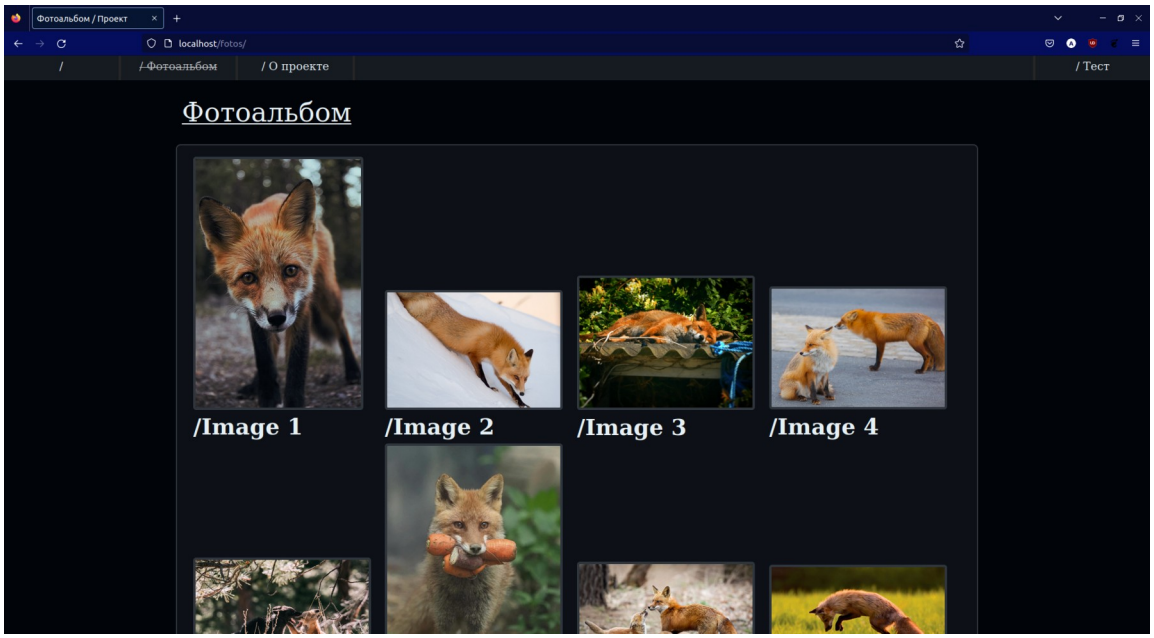

Рисунок 1 — Страница "Фотоальбом"

В странице «О проекте» в блоке «Интересы» контейнер с интересами заполняется с помощью скрипта (рисунок 2).

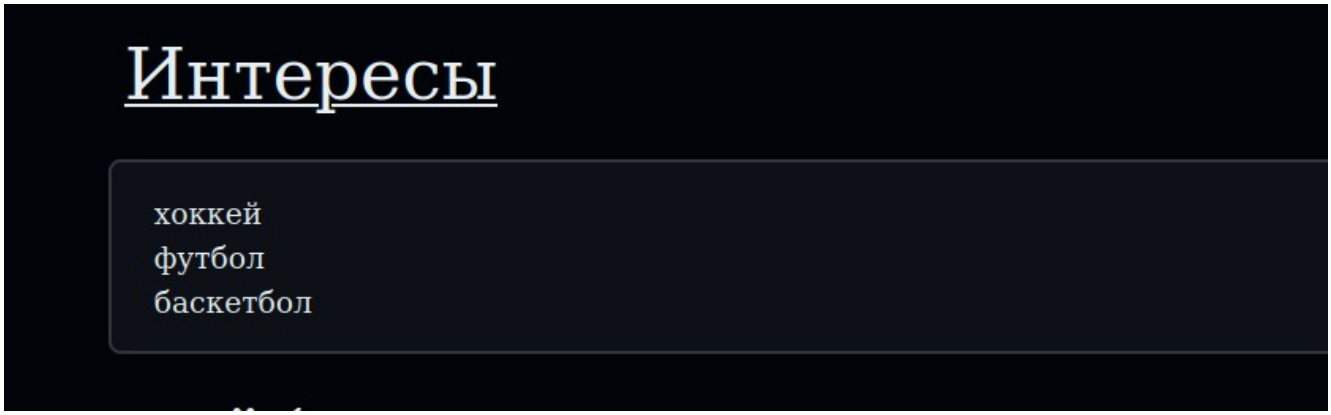

Рисунок 2 — Содержимое блока, заполненного с помощью скрипта

Рисунок 3 демонстрирует работу функции проверки заполненности полей формы на странице «Контакт», а рисунок 4 — на странице «Тест».

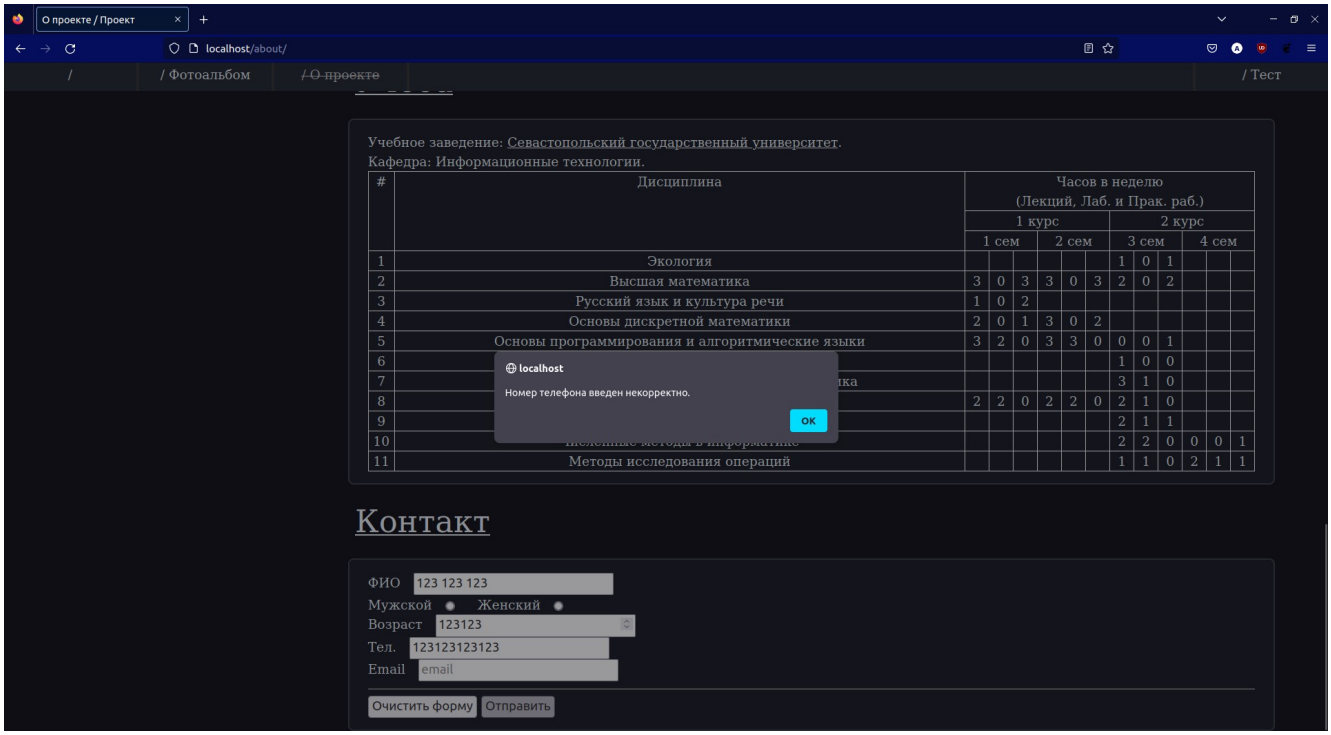

Рисунок 3 — Отображение ошибки на странице "Контакт"

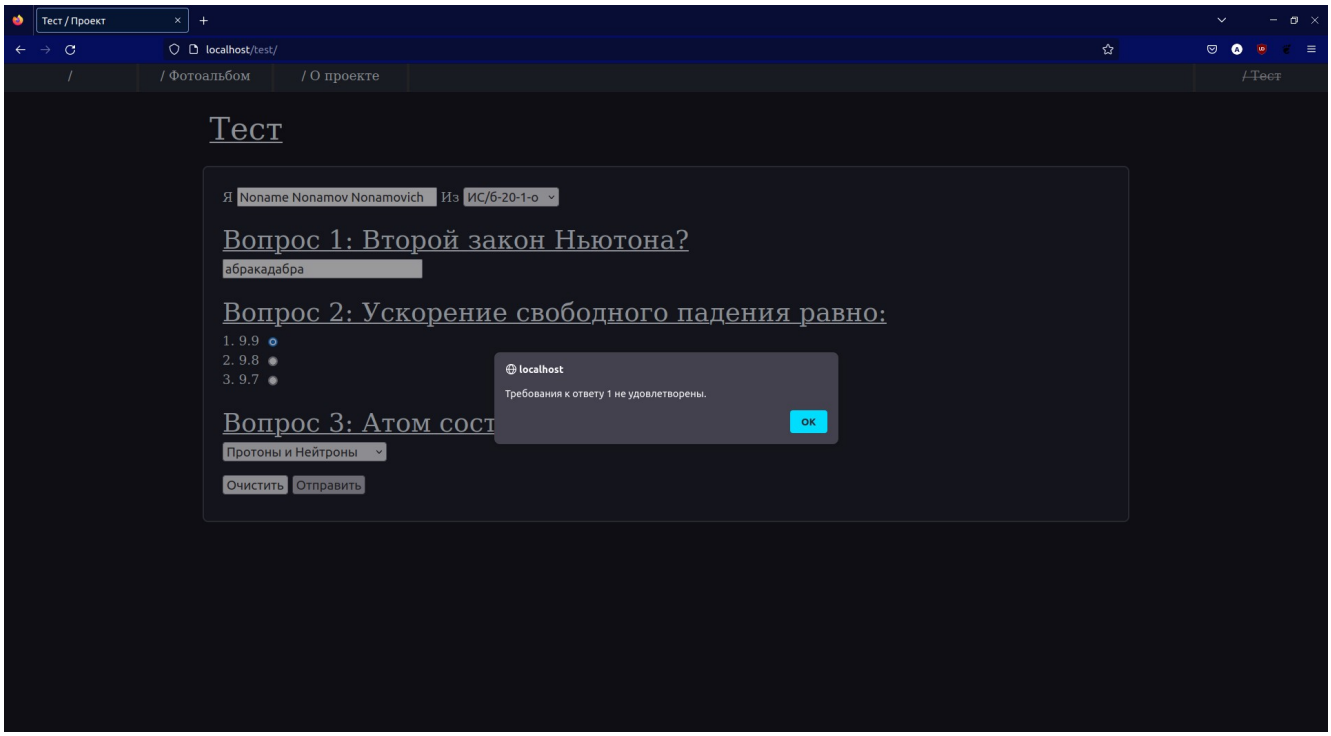

Рисунок 4 — Отображение ошибки на странице "Тест"

Вывод

При выполнении данной лабораторной работы были получены навыки написания скриптов для Web-ресурса с применением языка JavaScript.

Приложение Б

Листинг скриптов к лабораторной работе №2

Листинг скрипта /lib/script/HorriblePersonData.js

```
'use strict';

class HorriblePersonData
{
  constructor(x)
  {
    this.x = x;
  }

  isFIO()
  {
    return /\w{1,}\s\w{1,}\s\w{1,}/gm.test(this.x);
  }

  isTelephoneNumber()
  {
    return /\+[7|3][0-9]{9,11}/gm.test(this.x);
  }

  isAge()
  {
    return /[0-9]{1,3}/gm.test(this.x);
  }

  isEmail()
  {
    return /\w{1,}@\w{1,}\.\w{2,5}/gm.test(this.x);
  }

  isGroup()
  {
    return /\w{2,3}\/\w{1}-[0-9]{2}-[0-9]{1,2}-\w{1}/gmu.test(this.x);
  }
}
```

Листинг скрипта /lib/script/HorribleTestData.js

```
'use strict';

class HorribleTestData
{
  constructor(x)
  {
    this.x = x;
  }

  isFirstQuestionAnswer()
  {
    return /^[\w]+(\s+[\w]+){24}$/gmu.test(this.x);
  }
}
```

Листинг скрипта /lib/script/init_ContactDataFormValidator.js

```
function validateForm(e)
{
  let form = document.forms[0];
  function kill(event)
  {
    event.stopPropagation();
    return true;
  }
  function clearAndFocus(element)
```

```
  {
    element.value = "";
    element.focus();
  }

  let name = form.elements.name;
  let age = form.elements.age;
  let telnum = form.elements.telnum;
  let email = form.elements.email;

  if (!(new HorriblePersonData(name.value).isFIO()))
  {
    alert("ФИО введено некорректно.");
    clearAndFocus(name);
    return kill(e);
  }
  else if (!(new HorriblePersonData(age.value).isAge()))
  {
    alert("Возраст введен некорректно.");
    clearAndFocus(age);
    return kill(e);
  }
  else if (!(new HorriblePersonData(telnum.value).isTelephoneNumber()))
  {
    alert("Номер телефона введен некорректно.");
    clearAndFocus(telnum);
    return kill(e);
  }
  else if (!(new HorriblePersonData(email.value).isEmail()))
  {
    alert("Email введен некорректно.");
    clearAndFocus(email);
    return kill(e);
  }
  else
  {
    form.submit();
    return false;
  }

}

document.addEventListener("DOMContentLoaded", () =>
{
  document.getElementById("submitButton").addEventListener("click", validateForm, true);
});
```

Листинг скрипта /lib/script/init_PhotoTable.js

```
'use strict';

const kTableColumnCount = 4;

var photos = [
  new Photo("Image 1", "/lib/image/album/image1.png"),
  new Photo("Image 2", "/lib/image/album/image2.jpg"),
  new Photo("Image 3", "/lib/image/album/image3.jpg"),
  new Photo("Image 4", "/lib/image/album/image4.jpg"),
  new Photo("Image 5", "/lib/image/album/image5.jpg"),
  new Photo("Image 6", "/lib/image/album/image6.jpg"),
  new Photo("Image 7", "/lib/image/album/image7.jpg"),
  new Photo("Image 8", "/lib/image/album/image8.png"),
  new Photo("Image 9", "/lib/image/album/image9.jpg"),
  new Photo("Image 10", "/lib/image/album/image10.jpg"),
  new Photo("Image 11", "/lib/image/album/image11.png"),
  new Photo("Image 12", "/lib/image/album/image12.jpg"),
  new Photo("Image 13", "/lib/image/album/image13.jpg"),
  new Photo("Image 14", "/lib/image/album/image14.jpg"),
  new Photo("Image 15", "/lib/image/album/image15.jpg")
]

document.addEventListener("DOMContentLoaded", () =>
{
  let tableBody = document.getElementById("table-content");
```

```
  let preparedPhotos = photos.map((value, index, array) => { return new PhotoTableCell(value).asDOMString(); });

  for (let i = 0; i < kTableColumnCount; ++i)
  {
    let startPosition = i * kTableColumnCount;
    tableBody.append(
      new PhotoTableCells(
        preparedPhotos.slice(startPosition, startPosition + kTableColumnCount)
      ).joinIntoTableRow()
    );
  }
});
```

Листинг скрипта /lib/script/init_TestDataFormValidator.js

```
function validateForm(e)
{
  let form = document.forms[0];
  function kill(event)
  {
    event.stopPropagation();
    return true;
  }
  function clearAndFocus(element)
  {
    element.value = "";
    element.focus();
  }

  let name = form.elements.name;
  let answer1 = form.elements.answer1;

  if (!(new HorriblePersonData(name.value).isFIO()))
  {
    alert("ФИО введено некорректно.");
    clearAndFocus(name);
    return kill(e);
  }
  else if (!(new HorribleTestData(answer1.value).isFirstQuestionAnswer()))
  {
    alert("Требования к ответу 1 не удовлетворены.");
    clearAndFocus(answer1);
    return kill(e);
  }
  else
  {
    form.submit();
    return false;
  }

}

document.addEventListener("DOMContentLoaded", () =>
{
  document.getElementById("submitButton").addEventListener("click", validateForm, true);
});
```

Листинг скрипта /lib/script/inti_InterestsBlock.js

```
'use strict';

function representArgumentsAsListItems()
{
  let ret = []
  for (let i = 0; i < arguments.length; ++i)
  {
    ret.push(`<li>${arguments[i]}</li>`);
  }
  return ret;
}
```

```
document.addEventListener("DOMContentLoaded", () =>
{
  let interestsBlock = document.getElementById("interests-content");
  let listElement = document.createElement("ul");
  listElement.innerHTML = representArgumentsAsListItems("хоккей", "футбол", "баскетбол").join("");
  interestsBlock.append(listElement);
});
```

Листинг скрипта /lib/script/PhotoAdapters.js

```
'use strict';

class Photo
{
  constructor(title, url)
  {
    this.title = title;
    this.url = url;
  }
}
```

Листинг скрипта /lib/script/Photo.js

```
class PhotoTableCell
{
  constructor(photo)
  {
    this.photo = photo;
  }

  asDOMString()
  {
    return `
    <td>
      <a href="${this.photo.url}">
        <div class="image-frame">
          <div class="if-content"><img src="${this.photo.url}" alt="${this.photo.title}" title="$
{this.photo.title}"></div>
          <div class="if-title">/${this.photo.title}</div>
        </div>
      </a>
    </td>
    `;
  }
}

class PhotoTableCells
{
  constructor(photos)
  {
    this.photos = photos;
  }

  joinIntoTableRow()
  {
    let ret = document.createElement("tr");
    ret.innerHTML = this.photos.join("")
    return ret
  }
}
```