Лабораторна робота №13

Використання технології **AJAX(Asynchronous Javascript and XML)**.

**Мета роботи**: практичне застосування асинхронного завантаження файлів задопомогою технології AJAX

Для виконання подальших завдань необхідно використовувати інформацію з предметної області роботи №10 згідно свого варіанту, яку треба завантажувати з файлів різних форматів, які заздалегідь необхідно підготувати перед виконанням відповідного завдання

Завдання 1(завантаження з текстових файлів)

Модифікувати друге завдання роботи №10 таким чином, щоб дані для масивів-таблиць завантажувалися з текстових файлів та перетворювалися («парсилися») до відповідних об’єктів.

Програмний код:

Дидро;Монахиня;1678;230#

Дидро;Племянник Рамо;1699;600#

Митчел;Унесенные ветром;1936;789#

Ахматова;Избранное;1993;505#

Прутков;Полное собрание сочинений;1965;400#

Лондон;Белый клык;1977;900#

Ситель;Соло для позвоночника;2010;221#

Бутромеев;Древний мир;2000;333#

Гирченко;Законы Мэрфи;2006;254||

Иванов;ул. Токарева, 1;0506327533#

Петров;ул. Аиста, 13;050661335#

Сидоров;пер. Советский, 65;0501839016#

Конкин;ул. Пушкинская, 89;0663132344#

Либерман;ул. Тюринцева, 5;0985632478#

Трас;ул. Миргородская, 6;0503698521#

Мюллер;пр. Победы, 45;0634125412#

Авис;пер. Ленивый, 1;0665588687#

Симоненко;ул. Локка, 21;0984578211||

<!DOCTYPE html>

<html>

<head>

<title>lab13\_1</title>

<script type="text/javascript">

//cd /D d:/Sublime Text 3\Основы web- технологий\lab13\newapp\newapp (data1.js)

class Book {

#avtor;

#nameOfBook;

#year;

#price;

constructor(avtor, nameOfBook, year, price) {

this.avtor=avtor;

this.nameOfBook=nameOfBook;

this.year=year;

this.price=price;

}

toString() {

return `Avtor: ${this.avtor}; Book: ${this.nameOfBook}; Year: ${this.year}; Price: ${this.price}; `;

}

set setAvtor(value) {

this.avtor=value;

}

set setNameOfBook(value) {

this.nameOfBook=value;

}

set setYear(value) {

this.year=value;

}

set setPrice(value) {

this.price=value;

}

get getAvtor() {

return this.avtor;

}

get getNameOfBook() {

return this.nameOfBook;

}

get getYear() {

return this.year;

}

get getPrice() {

return this.price;

}

}

class Reader {

#nameOfReader;

#adres;

#tel;

constructor(nameOfReader, adres, tel) {

this.nameOfReader=nameOfReader;

this.adres=adres;

this.tel=tel;

}

toString() {

return `Reader: ${this.nameOfReader}; Adres: ${this.adres}; Tel.: ${this.tel}; `;

}

set setNameOfReader(value) {

this.nameOfReader=value;

}

set setAdres(value) {

this.adres=value;

}

set setTel(value) {

this.tel=value;

}

get getNameOfReader() {

return this.nameOfReader;

}

get getAdres() {

return this.adres;

}

get getTel() {

return this.tel;

}

}

class Note {

#reader;

#book;

#angabeDatum;

#zuruckgabeDatum;

#faktZD;

constructor(Reader, Book, angabeDatum, zuruckgabeDatum, faktZD) {

this.reader=Reader;

this.book=Book;

this.angabeDatum=angabeDatum;

this.zuruckgabeDatum=zuruckgabeDatum;

this.faktZD=faktZD;

}

toString() {

return `${this.reader.get} ${this.book.toString()} Angabedatum: ${this.angabeDatum}; Zurückgabedatum: ${this.zuruckgabeDatum}; Tatsächliches Zurückgabedatum: ${this.faktZD};`;

}

get getReader() {

return this.reader;

}

get getBook() {

return this.book;

}

get getAngabeDatum() {

return this.angabeDatum;

}

get getZuruckgabeDatum() {

return this.zuruckgabeDatum;

}

get getFaktZD() {

return this.faktZD;

}

}

function randomDate(start, end) {

var date = new Date(+start + Math.random() \* (end - start));

day=date.getDay();

month=date.getMonth();

year=date.getFullYear();

var newDate=`${year}.${month}.${day}`;

return newDate;

}

datas=[randomDate(new Date(2012, 0, 1), new Date()), randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date())];

function createTableBook(a, n,y,p) {

var tbody=document.getElementById('tableBook').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=a;

var td2=document.createElement('TD');

td2.innerHTML=n;

var td3=document.createElement('TD');

td3.innerHTML=y;

var td4=document.createElement('TD');

td4.innerHTML=p;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

row.appendChild(td4);

tbody.appendChild(row);

}

function createTableReader(n,a,t) {

var tbody=document.getElementById('tableReader').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=n;

var td2=document.createElement('TD');

td2.innerHTML=a;

var td3=document.createElement('TD');

td3.innerHTML=t;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

tbody.appendChild(row);

}

function createTableNote(r,b,a,z,f) {

var tbody=document.getElementById('tableNote').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=r;

var td2=document.createElement('TD');

td2.innerHTML=b;

var td3=document.createElement('TD');

td3.innerHTML=a;

var td4=document.createElement('TD');

td4.innerHTML=z;

var td5=document.createElement('TD');

td5.innerHTML=f;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

row.appendChild(td4);

row.appendChild(td5);

tbody.appendChild(row);

}

function createFormularBook(a,n,y,p) {

let forma=document.getElementsByClassName('formularBook')[0];

let bookk = document.createElement('fieldset');

bookk.appendChild(document.createElement('legend'));

bookk.getElementsByTagName("legend")[0].innerHTML='Buch';//innerHTML

bookk.appendChild(document.createElement('div'));

bookk.getElementsByTagName('div')[0].innerHTML=`Avtor: ${a};<br> Book: ${n};<br> Year: ${y};<br> Price: ${p};<br>`;

forma.appendChild(bookk);

}

function createFormularReader(n,a,t) {

let forma=document.getElementsByClassName('formularReader')[0];

let readerr=document.createElement('fieldset');

readerr.appendChild(document.createElement('legend'));

readerr.getElementsByTagName('legend')[0].innerHTML='Leser';

readerr.appendChild(document.createElement('div'));

readerr.getElementsByTagName('div')[0].innerHTML=`Reader: ${n};<br> Adres: ${a};<br> Tel.: ${t};<br>`;

forma.appendChild(readerr);

}

function createFormularNote(r,b,a,z,f) {

let forma=document.getElementsByClassName('formularNote')[0];

let notee=document.createElement('fieldset');

notee.appendChild(document.createElement('legend'));

notee.getElementsByTagName('legend')[0].innerHTML='Datenspeicherung';

notee.appendChild(document.createElement('div'));

notee.getElementsByTagName('div')[0].innerHTML=`${r}<br> ${b}<br> Angabedatum: ${a};<br> Zurückgabedatum: ${z};<br> Tatsächliches Zurückgabedatum: ${f};<br>`;

forma.appendChild(notee);

}

</script>

<style type="text/css">

.disabled {

display: none;

}

.abled {

display: block;

}

tr, td, th {

border: 3px inset #cccc00;

text-align: center;

}

tr {

background-color: yellow;

}

table {

margin-right: 15%;

margin-left: 15%;

}

th {

background-color: #e6e600;

}

body {

color: #808000;

font-size: 20px;

}

legend {

background-color: #ffffcc;

border: 1px outset #666600;

width: 30%;

}

fieldset {

border: 1px outset #666600;

background-color: #e6e600;

box-shadow: 0px 0px 5px 5px #cccc00;

}

select, option {

background-color: #ffffcc;

color: #808000;

font-size: 20px;

}

</style>

</head>

<body bgcolor="#ffff66">

<form>

<input type="radio" name="pack" id="radioTable" onclick="checkTable()"><label for="radioTable">Tabele</label><br>

<input type="radio" name="pack" id="radioFormular" onclick="checkTable()"><label for="radioFormular">Formular</label><br>

<select id="list" onchange="checkTable()">

<option disabled selected value="Options">Varianten</option>

<option id="listBook">Buch</option>

<option id="listReader">Leser</option>

<option id="listNote">Datenspeicherung</option>

</select>

<br>

</form>

<br>

<table id="tableBook" width="100%" class="disabled">

<tr><th>Avtor</th><th>Name of Book</th><th>Year</th><th>Price</th></tr>

</table>

<table id="tableReader" width="100%" class="disabled">

<tr><th>Name od Reader</th><th>Adres</th><th>Tel.</th></tr>

</table>

<table id="tableNote" width="100%" class="disabled">

<tr><th>Reader</th><th>Book</th><th>Angabedatum</th><th>Zuruck..datum</th><th>FaktZD</th></tr>

</table>

<form class="formularBook disabled" id="formBuch"></form>

<form class="formularReader disabled" id="formLeser"></form>

<form class="formularNote disabled" id="formDaten"></form>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script>

let arr1 = [];

let arr2=[];

let books = [];

let readers = [];

let notes=[];

var xhttp, textInf;

xhttp = new XMLHttpRequest();

xhttp.onload = function() {

if (xhttp.readyState == 4 && xhttp.status == 200) {

textInf = xhttp.response;

console.log(textInf);

let arrays = textInf.split('||');

arr1 = arrays[0].split('#');

arr2 = arrays[1].split('#');

for(let i=0; i < arr1.length; i++){

let obj = arr1[i].split(';');

books.push(new Book(obj[0],obj[1],parseInt(obj[2]), parseInt(obj[3])));

}

for(let i=0; i < arr2.length; i++){

let obj = arr2[i].split(';');

readers.push(new Reader(obj[0],obj[1],obj[2]));

}

books.forEach(book=>{

console.log(book.toString())

});

readers.forEach(reader=>{

console.log(reader.toString())

});

notes=[new Note(books[0], readers[0],datas[0],datas[1],datas[2]), new Note (books[1], readers[1], datas[3], datas[4], datas[5]), new Note (books[2], readers[2], datas[6], datas[7], datas[8]),new Note (books[3], readers[3], datas[9], datas[10], datas[11]) ];

notes.forEach(note=>{

console.log(note.toString())

});

for(let b of books) {

createTableBook(b.getAvtor, b.getNameOfBook, b.getYear, b.getPrice);

}

for(let r of readers) {

createTableReader(r.getNameOfReader, r.getAdres, r.getTel);

}

for(let n of notes) {

createTableNote(n.getReader, n.getBook, n.getAngabeDatum, n.getZuruckgabeDatum, n.getFaktZD);

}

//formulars

for(let b of books) {

createFormularBook(b.getAvtor, b.getNameOfBook, b.getYear, b.getPrice);

}

for(let r of readers) {

createFormularReader(r.getNameOfReader, r.getAdres, r.getTel);

}

for(let n of notes) {

createFormularNote(n.getReader, n.getBook, n.getAngabeDatum, n.getZuruckgabeDatum, n.getFaktZD);

}

}

};

xhttp.open("GET", "http://localhost:8080");

xhttp.send();

//tables

function checkTable() {

let buch=document.getElementById('listBook');

let leser=document.getElementById('listReader');

let daten=document.getElementById('listNote');

if(radioTable.checked){

formBuch.className='formularBook disabled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote disabled';

if(buch.selected){

tableBook.className='abled';

tableReader.className='disabled';

tableNote.className='disabled';

console.log("buchTabele");

}

else if(leser.selected){

tableReader.className='abled';

tableBook.className='disabled';

tableNote.className='disabled';

console.log("leserTabele checked");

} else if(daten.selected){

tableReader.className='disabled';

tableBook.className='disabled';

tableNote.className='abled';

console.log("datenTabele checked");

}

} else if(radioFormular.checked) {

tableReader.className='disabled';

tableBook.className='disabled';

tableNote.className='disabled';

if(buch.selected) {

console.log('radioBuch checked');

formBuch.className='formularBook abled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote disabled';

} else if(leser.selected) {

formBuch.className='formularBook disabled';

formLeser.className='formularReader abled';

formDaten.className='formularNote disabled';

} else if(daten.selected) {

formBuch.className='formularBook disabled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote abled';

}

}

}

</script>

</body>

</html>

Завдання 2(завантаження з XML – файлів, теговий формат)

Модифікувати друге завдання роботи №10 таким чином, щоб дані для масивів-таблиць завантажувалися з XML файлів, які мають теговий формат, та перетворювалися («парсилися») до відповідних об’єктів.

Програмний код:

<?xml version="1.0"?>

<Data>

<Book>

<el>

<avtor>Дидро</avtor>

<nameOfBook>Монахиня</nameOfBook>

<year>1678</year>

<price>230</price>

</el>

<el>

<avtor>Дидро</avtor>

<nameOfBook>Племянник Рамо</nameOfBook>

<year>1699</year>

<price>600</price>

</el>

<el>

<avtor>Митчел</avtor>

<nameOfBook>Унесенные ветром</nameOfBook>

<year>1936</year>

<price>789</price>

</el>

<el>

<avtor>Ахматова</avtor>

<nameOfBook>Избранное</nameOfBook>

<year>1993</year>

<price>505</price>

</el>

<el>

<avtor>Прутков</avtor>

<nameOfBook>Полное собрание сочинений</nameOfBook>

<year>1965</year>

<price>400</price>

</el>

<el>

<avtor>Лондон</avtor>

<nameOfBook>Белый клык</nameOfBook>

<year>1997</year>

<price>900</price>

</el>

<el>

<avtor>Ситель</avtor>

<nameOfBook>Соло для позвоночника</nameOfBook>

<year>2010</year>

<price>221</price>

</el>

<el>

<avtor>Бутромеев</avtor>

<nameOfBook>Древний мир</nameOfBook>

<year>2000</year>

<price>333</price>

</el>

<el>

<avtor>Гирченко</avtor>

<nameOfBook>Законы Мэрфи</nameOfBook>

<year>2006</year>

<price>254</price>

</el>

</Book>

<Reader>

<el>

<nameOfReader>Иванов</nameOfReader>

<adres>улю ТОкарева, 1</adres>

<tel>0506327533</tel>

</el>

<el>

<nameOfReader>Петров</nameOfReader>

<adres>ул. Аиста, 13</adres>

<tel>0506631335</tel>

</el>

<el>

<nameOfReader>Сидоров</nameOfReader>

<adres>пер. Советский, 65</adres>

<tel>0501839016</tel>

</el>

<el>

<nameOfReader>Конкин</nameOfReader>

<adres>ул. Пушкинская, 89</adres>

<tel>0663132344</tel>

</el>

<el>

<nameOfReader>Либерман</nameOfReader>

<adres>ул. Тюринцева, 5</adres>

<tel>985632478</tel>

</el>

<el>

<nameOfReader>Трас</nameOfReader>

<adres>ул. Миргородская, 6</adres>

<tel>0503698521</tel>

</el>

<el>

<nameOfReader>Мюллер</nameOfReader>

<adres>пр. Победы, 45</adres>

<tel>0634125412</tel>

</el>

<el>

<nameOfReader>Авис</nameOfReader>

<adres>пер. Ленивый, 1</adres>

<tel>0665588687</tel>

</el>

<el>

<nameOfReader>Симоненко</nameOfReader>

<adres>ул. Локка, 21</adres>

<tel>0984578211</tel>

</el>

</Reader>

</Data>

<!DOCTYPE html>

<html>

<head>

<title>lab13\_2</title>

<script type="text/javascript">

class Book {

#avtor;

#nameOfBook;

#year;

#price;

constructor(avtor, nameOfBook, year, price) {

this.avtor=avtor;

this.nameOfBook=nameOfBook;

this.year=year;

this.price=price;

}

toString() {

return `Avtor: ${this.avtor}; Book: ${this.nameOfBook}; Year: ${this.year}; Price: ${this.price}; `;

}

set setAvtor(value) {

this.avtor=value;

}

set setNameOfBook(value) {

this.nameOfBook=value;

}

set setYear(value) {

this.year=value;

}

set setPrice(value) {

this.price=value;

}

get getAvtor() {

return this.avtor;

}

get getNameOfBook() {

return this.nameOfBook;

}

get getYear() {

return this.year;

}

get getPrice() {

return this.price;

}

}

class Reader {

#nameOfReader;

#adres;

#tel;

constructor(nameOfReader, adres, tel) {

this.nameOfReader=nameOfReader;

this.adres=adres;

this.tel=tel;

}

toString() {

return `Reader: ${this.nameOfReader}; Adres: ${this.adres}; Tel.: ${this.tel}; `;

}

set setNameOfReader(value) {

this.nameOfReader=value;

}

set setAdres(value) {

this.adres=value;

}

set setTel(value) {

this.tel=value;

}

get getNameOfReader() {

return this.nameOfReader;

}

get getAdres() {

return this.adres;

}

get getTel() {

return this.tel;

}

}

class Note {

#reader;

#book;

#angabeDatum;

#zuruckgabeDatum;

#faktZD;

constructor(Reader, Book, angabeDatum, zuruckgabeDatum, faktZD) {

this.reader=Reader;

this.book=Book;

this.angabeDatum=angabeDatum;

this.zuruckgabeDatum=zuruckgabeDatum;

this.faktZD=faktZD;

}

toString() {

return `${this.reader.get} ${this.book.toString()} Angabedatum: ${this.angabeDatum}; Zurückgabedatum: ${this.zuruckgabeDatum}; Tatsächliches Zurückgabedatum: ${this.faktZD};`;

}

get getReader() {

return this.reader;

}

get getBook() {

return this.book;

}

get getAngabeDatum() {

return this.angabeDatum;

}

get getZuruckgabeDatum() {

return this.zuruckgabeDatum;

}

get getFaktZD() {

return this.faktZD;

}

}

function randomDate(start, end) {

var date = new Date(+start + Math.random() \* (end - start));

day=date.getDay();

month=date.getMonth();

year=date.getFullYear();

var newDate=`${year}.${month}.${day}`;

return newDate;

}

datas=[randomDate(new Date(2012, 0, 1), new Date()), randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date())];

function createTableBook(a, n,y,p) {

var tbody=document.getElementById('tableBook').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=a;

var td2=document.createElement('TD');

td2.innerHTML=n;

var td3=document.createElement('TD');

td3.innerHTML=y;

var td4=document.createElement('TD');

td4.innerHTML=p;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

row.appendChild(td4);

tbody.appendChild(row);

}

function createTableReader(n,a,t) {

var tbody=document.getElementById('tableReader').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=n;

var td2=document.createElement('TD');

td2.innerHTML=a;

var td3=document.createElement('TD');

td3.innerHTML=t;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

tbody.appendChild(row);

}

function createTableNote(r,b,a,z,f) {

var tbody=document.getElementById('tableNote').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=r;

var td2=document.createElement('TD');

td2.innerHTML=b;

var td3=document.createElement('TD');

td3.innerHTML=a;

var td4=document.createElement('TD');

td4.innerHTML=z;

var td5=document.createElement('TD');

td5.innerHTML=f;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

row.appendChild(td4);

row.appendChild(td5);

tbody.appendChild(row);

}

function createFormularBook(a,n,y,p) {

let forma=document.getElementsByClassName('formularBook')[0];

let bookk = document.createElement('fieldset');

bookk.appendChild(document.createElement('legend'));

bookk.getElementsByTagName("legend")[0].innerHTML='Buch';//innerHTML

bookk.appendChild(document.createElement('div'));

bookk.getElementsByTagName('div')[0].innerHTML=`Avtor: ${a};<br> Book: ${n};<br> Year: ${y};<br> Price: ${p};<br>`;

forma.appendChild(bookk);

}

function createFormularReader(n,a,t) {

let forma=document.getElementsByClassName('formularReader')[0];

let readerr=document.createElement('fieldset');

readerr.appendChild(document.createElement('legend'));

readerr.getElementsByTagName('legend')[0].innerHTML='Leser';

readerr.appendChild(document.createElement('div'));

readerr.getElementsByTagName('div')[0].innerHTML=`Reader: ${n};<br> Adres: ${a};<br> Tel.: ${t};<br>`;

forma.appendChild(readerr);

}

function createFormularNote(r,b,a,z,f) {

let forma=document.getElementsByClassName('formularNote')[0];

let notee=document.createElement('fieldset');

notee.appendChild(document.createElement('legend'));

notee.getElementsByTagName('legend')[0].innerHTML='Datenspeicherung';

notee.appendChild(document.createElement('div'));

notee.getElementsByTagName('div')[0].innerHTML=`${r}<br> ${b}<br> Angabedatum: ${a};<br> Zurückgabedatum: ${z};<br> Tatsächliches Zurückgabedatum: ${f};<br>`;

forma.appendChild(notee);

}

</script>

<style type="text/css">

.disabled {

display: none;

}

.abled {

display: block;

}

tr, td, th {

border: 3px inset #cccc00;

text-align: center;

}

tr {

background-color: yellow;

}

table {

margin-right: 15%;

margin-left: 15%;

}

th {

background-color: #e6e600;

}

body {

color: #808000;

font-size: 20px;

}

legend {

background-color: #ffffcc;

border: 1px outset #666600;

width: 30%;

}

fieldset {

border: 1px outset #666600;

background-color: #e6e600;

box-shadow: 0px 0px 5px 5px #cccc00;

}

select, option {

background-color: #ffffcc;

color: #808000;

font-size: 20px;

}

</style>

</head>

<body bgcolor="#ffff66">

<form>

<input type="radio" name="pack" id="radioTable" onclick="checkTable()"><label for="radioTable">Tabele</label><br>

<input type="radio" name="pack" id="radioFormular" onclick="checkTable()"><label for="radioFormular">Formular</label><br>

<select id="list" onchange="checkTable()">

<option disabled selected value="Options">Varianten</option>

<option id="listBook">Buch</option>

<option id="listReader">Leser</option>

<option id="listNote">Datenspeicherung</option>

</select>

<br>

</form>

<br>

<table id="tableBook" width="100%" class="disabled">

<tr><th>Avtor</th><th>Name of Book</th><th>Year</th><th>Price</th></tr>

</table>

<table id="tableReader" width="100%" class="disabled">

<tr><th>Name od Reader</th><th>Adres</th><th>Tel.</th></tr>

</table>

<table id="tableNote" width="100%" class="disabled">

<tr><th>Reader</th><th>Book</th><th>Angabedatum</th><th>Zuruck..datum</th><th>FaktZD</th></tr>

</table>

<form class="formularBook disabled" id="formBuch"></form>

<form class="formularReader disabled" id="formLeser"></form>

<form class="formularNote disabled" id="formDaten"></form>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script>

let arr1 = [];

let arr2=[];

let books = [];

let readers = [];

let notes=[];

var xhttp, textInf;

xhttp = new XMLHttpRequest();

xhttp.onload = function() {

if (xhttp.readyState == 4 && xhttp.status == 200) {

textInf = xhttp.responseXML;

arr1 = textInf.getElementsByTagName("Book")[0].children;

arr2 = textInf.getElementsByTagName("Reader")[0].children;

for(let i=0; i < arr1.length; i++){

let avtor = arr1[i].children[0].innerHTML;

let nameOfBook = arr1[i].children[1].innerHTML;

let year = parseInt(arr1[i].children[2].innerHTML);

let price=parseInt(arr1[i].children[3].innerHTML);

books.push(new Book(avtor, nameOfBook, year, price));

}

for(let i=0; i < arr2.length; i++){

let nameOfReader = arr2[i].children[0].innerHTML;

let adres = arr2[i].children[1].innerHTML;

let tel =arr2[i].children[2].innerHTML;

readers.push(new Reader(nameOfReader, adres, tel));

}

books.forEach(book=>{

console.log(book.toString())

});

readers.forEach(reader=>{

console.log(reader.toString())

});

notes=[new Note(books[0], readers[0],datas[0],datas[1],datas[2]), new Note (books[1], readers[1], datas[3], datas[4], datas[5]), new Note (books[2], readers[2], datas[6], datas[7], datas[8]),new Note (books[3], readers[3], datas[9], datas[10], datas[11]) ];

notes.forEach(note=>{

console.log(note.toString())

});

for(let b of books) {

createTableBook(b.getAvtor, b.getNameOfBook, b.getYear, b.getPrice);

}

for(let r of readers) {

createTableReader(r.getNameOfReader, r.getAdres, r.getTel);

}

for(let n of notes) {

createTableNote(n.getReader, n.getBook, n.getAngabeDatum, n.getZuruckgabeDatum, n.getFaktZD);

}

//formulars

for(let b of books) {

createFormularBook(b.getAvtor, b.getNameOfBook, b.getYear, b.getPrice);

}

for(let r of readers) {

createFormularReader(r.getNameOfReader, r.getAdres, r.getTel);

}

for(let n of notes) {

createFormularNote(n.getReader, n.getBook, n.getAngabeDatum, n.getZuruckgabeDatum, n.getFaktZD);

}

}

};

xhttp.open("GET", "http://localhost:8080");

xhttp.send();

//tables

function checkTable() {

let buch=document.getElementById('listBook');

let leser=document.getElementById('listReader');

let daten=document.getElementById('listNote');

if(radioTable.checked){

formBuch.className='formularBook disabled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote disabled';

if(buch.selected){

tableBook.className='abled';

tableReader.className='disabled';

tableNote.className='disabled';

console.log("buchTabele");

}

else if(leser.selected){

tableReader.className='abled';

tableBook.className='disabled';

tableNote.className='disabled';

console.log("leserTabele checked");

} else if(daten.selected){

tableReader.className='disabled';

tableBook.className='disabled';

tableNote.className='abled';

console.log("datenTabele checked");

}

} else if(radioFormular.checked) {

tableReader.className='disabled';

tableBook.className='disabled';

tableNote.className='disabled';

if(buch.selected) {

console.log('radioBuch checked');

formBuch.className='formularBook abled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote disabled';

} else if(leser.selected) {

formBuch.className='formularBook disabled';

formLeser.className='formularReader abled';

formDaten.className='formularNote disabled';

} else if(daten.selected) {

formBuch.className='formularBook disabled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote abled';

}

}

}

</script>

</body>

</html>

Завдання 3(завантаження з XML – файлів, атрибутивний формат)

Модифікувати друге завдання роботи №10 таким чином, щоб дані для масивів-таблиць завантажувалися з XML файлів, які мають атрибутивний формат, та перетворювалися («парсилися») до відповідних об’єктів.

Програмний код:

<?xml version="1.0"?>

<Data>

<Book>

<el

avtor="Дидро"

nameOfBook="Монахиня"

year="1678"

price="230"

>

</el>

<el

avtor="Дидро"

nameOfBook="Племянник Рамо"

year="1699"

price="600"

>

</el>

<el

avtor="Митчелл"

nameOfBook="Унесенные ветром"

year="1936"

price="789"

>

</el>

<el

avtor="Ахматова"

nameOfBook="Избранное"

year="1993"

price="505"

>

</el>

<el

avtor="Прутков"

nameOfBook="Полное собрание сочинений"

year="1965"

price="400"

>

</el>

<el

avtor="Лондон"

nameOfBook="Белый клык"

year="1997"

price="900"

>

</el>

<el

avtor="Ситель"

nameOfBook="Соло для позвоночника"

year="2010"

price="221"

>

</el>

<el

avtor="Бутромеев"

nameOfBook="Древний мир"

year="2000"

price="333"

>

</el>

<el

avtor="Гирченко"

nameOfBook="Законы Мэрфи"

year="2006"

price="254"

>

</el>

</Book>

<Reader>

<el

nameOfReader="Иванов"

adres="ул. Тoкарева, 1"

tel="0506327533"

>

</el>

<el

nameOfReader="Петров"

adres="ул. Аиста, 13"

tel="0506631335"

>

</el>

<el

nameOfReader="Сидоров"

adres="пер. Советский, 65"

tel="0501839016"

>

</el>

<el

nameOfReader="Конкин"

adres="ул. Пушкинская, 89"

tel="0663132344"

>

</el>

<el

nameOfReader="Либерман"

adres="ул. Тюринцева, 5"

tel="0985632478"

>

</el>

<el

nameOfReader="Трас"

adres="ул. Миргородская, 6"

tel="0503698521"

>

</el>

<el

nameOfReader="Мюллер"

adres="пр. Победы, 45"

tel="0634125412"

>

</el>

<el

nameOfReader="Авис"

adres="пер. Ленивый, 1"

tel="0665588687"

>

</el>

<el

nameOfReader="Симоненко"

adres="ул. Локка, 21"

tel="0984578211"

>

</el>

</Reader>

</Data>

<!DOCTYPE html>

<html>

<head>

<title>lab13\_3</title>

<script type="text/javascript">

class Book {

#avtor;

#nameOfBook;

#year;

#price;

constructor(avtor, nameOfBook, year, price) {

this.avtor=avtor;

this.nameOfBook=nameOfBook;

this.year=year;

this.price=price;

}

toString() {

return `Avtor: ${this.avtor}; Book: ${this.nameOfBook}; Year: ${this.year}; Price: ${this.price}; `;

}

set setAvtor(value) {

this.avtor=value;

}

set setNameOfBook(value) {

this.nameOfBook=value;

}

set setYear(value) {

this.year=value;

}

set setPrice(value) {

this.price=value;

}

get getAvtor() {

return this.avtor;

}

get getNameOfBook() {

return this.nameOfBook;

}

get getYear() {

return this.year;

}

get getPrice() {

return this.price;

}

}

class Reader {

#nameOfReader;

#adres;

#tel;

constructor(nameOfReader, adres, tel) {

this.nameOfReader=nameOfReader;

this.adres=adres;

this.tel=tel;

}

toString() {

return `Reader: ${this.nameOfReader}; Adres: ${this.adres}; Tel.: ${this.tel}; `;

}

set setNameOfReader(value) {

this.nameOfReader=value;

}

set setAdres(value) {

this.adres=value;

}

set setTel(value) {

this.tel=value;

}

get getNameOfReader() {

return this.nameOfReader;

}

get getAdres() {

return this.adres;

}

get getTel() {

return this.tel;

}

}

class Note {

#reader;

#book;

#angabeDatum;

#zuruckgabeDatum;

#faktZD;

constructor(Reader, Book, angabeDatum, zuruckgabeDatum, faktZD) {

this.reader=Reader;

this.book=Book;

this.angabeDatum=angabeDatum;

this.zuruckgabeDatum=zuruckgabeDatum;

this.faktZD=faktZD;

}

toString() {

return `${this.reader.get} ${this.book.toString()} Angabedatum: ${this.angabeDatum}; Zurückgabedatum: ${this.zuruckgabeDatum}; Tatsächliches Zurückgabedatum: ${this.faktZD};`;

}

get getReader() {

return this.reader;

}

get getBook() {

return this.book;

}

get getAngabeDatum() {

return this.angabeDatum;

}

get getZuruckgabeDatum() {

return this.zuruckgabeDatum;

}

get getFaktZD() {

return this.faktZD;

}

}

function randomDate(start, end) {

var date = new Date(+start + Math.random() \* (end - start));

day=date.getDay();

month=date.getMonth();

year=date.getFullYear();

var newDate=`${year}.${month}.${day}`;

return newDate;

}

datas=[randomDate(new Date(2012, 0, 1), new Date()), randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date())];

function createTableBook(a, n,y,p) {

var tbody=document.getElementById('tableBook').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=a;

var td2=document.createElement('TD');

td2.innerHTML=n;

var td3=document.createElement('TD');

td3.innerHTML=y;

var td4=document.createElement('TD');

td4.innerHTML=p;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

row.appendChild(td4);

tbody.appendChild(row);

}

function createTableReader(n,a,t) {

var tbody=document.getElementById('tableReader').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=n;

var td2=document.createElement('TD');

td2.innerHTML=a;

var td3=document.createElement('TD');

td3.innerHTML=t;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

tbody.appendChild(row);

}

function createTableNote(r,b,a,z,f) {

var tbody=document.getElementById('tableNote').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=r;

var td2=document.createElement('TD');

td2.innerHTML=b;

var td3=document.createElement('TD');

td3.innerHTML=a;

var td4=document.createElement('TD');

td4.innerHTML=z;

var td5=document.createElement('TD');

td5.innerHTML=f;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

row.appendChild(td4);

row.appendChild(td5);

tbody.appendChild(row);

}

function createFormularBook(a,n,y,p) {

let forma=document.getElementsByClassName('formularBook')[0];

let bookk = document.createElement('fieldset');

bookk.appendChild(document.createElement('legend'));

bookk.getElementsByTagName("legend")[0].innerHTML='Buch';//innerHTML

bookk.appendChild(document.createElement('div'));

bookk.getElementsByTagName('div')[0].innerHTML=`Avtor: ${a};<br> Book: ${n};<br> Year: ${y};<br> Price: ${p};<br>`;

forma.appendChild(bookk);

}

function createFormularReader(n,a,t) {

let forma=document.getElementsByClassName('formularReader')[0];

let readerr=document.createElement('fieldset');

readerr.appendChild(document.createElement('legend'));

readerr.getElementsByTagName('legend')[0].innerHTML='Leser';

readerr.appendChild(document.createElement('div'));

readerr.getElementsByTagName('div')[0].innerHTML=`Reader: ${n};<br> Adres: ${a};<br> Tel.: ${t};<br>`;

forma.appendChild(readerr);

}

function createFormularNote(r,b,a,z,f) {

let forma=document.getElementsByClassName('formularNote')[0];

let notee=document.createElement('fieldset');

notee.appendChild(document.createElement('legend'));

notee.getElementsByTagName('legend')[0].innerHTML='Datenspeicherung';

notee.appendChild(document.createElement('div'));

notee.getElementsByTagName('div')[0].innerHTML=`${r}<br> ${b}<br> Angabedatum: ${a};<br> Zurückgabedatum: ${z};<br> Tatsächliches Zurückgabedatum: ${f};<br>`;

forma.appendChild(notee);

}

</script>

<style type="text/css">

.disabled {

display: none;

}

.abled {

display: block;

}

tr, td, th {

border: 3px inset #cccc00;

text-align: center;

}

tr {

background-color: yellow;

}

table {

margin-right: 15%;

margin-left: 15%;

}

th {

background-color: #e6e600;

}

body {

color: #808000;

font-size: 20px;

}

legend {

background-color: #ffffcc;

border: 1px outset #666600;

width: 30%;

}

fieldset {

border: 1px outset #666600;

background-color: #e6e600;

box-shadow: 0px 0px 5px 5px #cccc00;

}

select, option {

background-color: #ffffcc;

color: #808000;

font-size: 20px;

}

</style>

</head>

<body bgcolor="#ffff66">

<form>

<input type="radio" name="pack" id="radioTable" onclick="checkTable()"><label for="radioTable">Tabele</label><br>

<input type="radio" name="pack" id="radioFormular" onclick="checkTable()"><label for="radioFormular">Formular</label><br>

<select id="list" onchange="checkTable()">

<option disabled selected value="Options">Varianten</option>

<option id="listBook">Buch</option>

<option id="listReader">Leser</option>

<option id="listNote">Datenspeicherung</option>

</select>

<br>

</form>

<br>

<table id="tableBook" width="100%" class="disabled">

<tr><th>Avtor</th><th>Name of Book</th><th>Year</th><th>Price</th></tr>

</table>

<table id="tableReader" width="100%" class="disabled">

<tr><th>Name od Reader</th><th>Adres</th><th>Tel.</th></tr>

</table>

<table id="tableNote" width="100%" class="disabled">

<tr><th>Reader</th><th>Book</th><th>Angabedatum</th><th>Zuruck..datum</th><th>FaktZD</th></tr>

</table>

<form class="formularBook disabled" id="formBuch"></form>

<form class="formularReader disabled" id="formLeser"></form>

<form class="formularNote disabled" id="formDaten"></form>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script>

let arr1 = [];

let arr2=[];

let books = [];

let readers = [];

let notes=[];

var xhttp, textInf;

xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (xhttp.readyState == 4 && xhttp.status == 200) {

textInf = xhttp.responseXML;

arr1 = textInf.getElementsByTagName("Book")[0].children;

arr2 = textInf.getElementsByTagName("Reader")[0].children;

for(let i=0; i < arr1.length; i++){

let avtor = arr1[i].getAttribute('avtor');

let nameOfBook = arr1[i].getAttribute('nameOfBook');

let year = parseInt(arr1[i].getAttribute('year'));

let price=parseInt(arr1[i].getAttribute('price'));

books.push(new Book(avtor, nameOfBook, year, price));

}

for(let i=0; i < arr2.length; i++){

let nameOfReader = arr2[i].getAttribute('nameOfReader');

let adres = arr2[i].getAttribute('adres');

let tel =arr2[i].getAttribute('tel');

readers.push(new Reader(nameOfReader, adres, tel));

}

books.forEach(book=>{

console.log(book.toString())

});

readers.forEach(reader=>{

console.log(reader.toString())

});

notes=[new Note(books[0], readers[0],datas[0],datas[1],datas[2]), new Note (books[1], readers[1], datas[3], datas[4], datas[5]), new Note (books[2], readers[2], datas[6], datas[7], datas[8]),new Note (books[3], readers[3], datas[9], datas[10], datas[11]) ];

notes.forEach(note=>{

console.log(note.toString())

});

for(let b of books) {

createTableBook(b.getAvtor, b.getNameOfBook, b.getYear, b.getPrice);

}

for(let r of readers) {

createTableReader(r.getNameOfReader, r.getAdres, r.getTel);

}

for(let n of notes) {

createTableNote(n.getReader, n.getBook, n.getAngabeDatum, n.getZuruckgabeDatum, n.getFaktZD);

}

//formulars

for(let b of books) {

createFormularBook(b.getAvtor, b.getNameOfBook, b.getYear, b.getPrice);

}

for(let r of readers) {

createFormularReader(r.getNameOfReader, r.getAdres, r.getTel);

}

for(let n of notes) {

createFormularNote(n.getReader, n.getBook, n.getAngabeDatum, n.getZuruckgabeDatum, n.getFaktZD);

}

}

};

xhttp.open("GET", "http://localhost:8080");

xhttp.send();

//tables

function checkTable() {

let buch=document.getElementById('listBook');

let leser=document.getElementById('listReader');

let daten=document.getElementById('listNote');

if(radioTable.checked){

formBuch.className='formularBook disabled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote disabled';

if(buch.selected){

tableBook.className='abled';

tableReader.className='disabled';

tableNote.className='disabled';

console.log("buchTabele");

}

else if(leser.selected){

tableReader.className='abled';

tableBook.className='disabled';

tableNote.className='disabled';

console.log("leserTabele checked");

} else if(daten.selected){

tableReader.className='disabled';

tableBook.className='disabled';

tableNote.className='abled';

console.log("datenTabele checked");

}

} else if(radioFormular.checked) {

tableReader.className='disabled';

tableBook.className='disabled';

tableNote.className='disabled';

if(buch.selected) {

console.log('radioBuch checked');

formBuch.className='formularBook abled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote disabled';

} else if(leser.selected) {

formBuch.className='formularBook disabled';

formLeser.className='formularReader abled';

formDaten.className='formularNote disabled';

} else if(daten.selected) {

formBuch.className='formularBook disabled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote abled';

}

}

}

</script>

</body>

</html>

Завдання 4(завантаження з XML – файлів, змішаний формат)

Модифікувати друге завдання роботи №10 таким чином, щоб дані для

масивів-таблиць завантажувалися з XML файлів, які мають змішаний

(інформація може зберігатися як в тегах так і в атрибутах тегів) формат, та

перетворювалися («парсилися») до відповідних об’єктів. Структуру такого

XML файлу сформувати самостійно таким чином, щоб половина полів

кожного запису за бажанням розробника мала теговий формат, а інша

половина - атрибутивний

Програмний код:

<?xml version="1.0"?>

<Data>

<Book>

<el

avtor="Дидро"

nameOfBook="Монахиня"

>

<year>1678</year>

<price>230</price>

</el>

<el

avtor="Дидро"

nameOfBook="Племянник Рамо"

>

<year>1699</year>

<price>600</price>

</el>

<el

avtor="Митчелл"

nameOfBook="Унесенные ветром"

>

<year>1936</year>

<price>789</price>

</el>

<el

avtor="Ахматова"

nameOfBook="Избранное"

>

<year>1993</year>

<price>505</price>

</el>

<el

avtor="Прутков"

nameOfBook="Полное собрание сочинений"

>

<year>1965</year>

<price>400</price>

</el>

<el

avtor="Лондон"

nameOfBook="Белый клык"

>

<year>1997</year>

<price>900</price>

</el>

<el

avtor="Ситель"

nameOfBook="Соло для позвоночника"

>

<year>2010</year>

<price>221</price>

</el>

<el

avtor="Бутромеев"

nameOfBook="Древний мир"

>

<year>2000</year>

<price>333</price>

</el>

<el

avtor="Гирченко"

nameOfBook="Законы Мэрфи"

>

<year>2006</year>

<price>254</price>

</el>

</Book>

<Reader>

<el

nameOfReader="Иванов"

>

<adres>улю ТОкарева, 1</adres>

<tel>0506327533</tel>

</el>

<el

nameOfReader="Петров"

>

<adres>ул. Аиста, 13</adres>

<tel>0506631335</tel>

</el>

<el

nameOfReader="Сидоров"

>

<adres>пер. Советский, 65</adres>

<tel>0501839016</tel>

</el>

<el

nameOfReader="Конкин"

>

<adres>ул. Пушкинская, 89</adres>

<tel>0663132344</tel>

</el>

<el

nameOfReader="Либерман"

>

<adres>ул. Тюринцева, 5</adres>

<tel>985632478</tel>

</el>

<el

nameOfReader="Трас"

>

<adres>ул. Миргородская, 6</adres>

<tel>0503698521</tel>

</el>

<el

nameOfReader="Мюллер"

>

<adres>пр. Победы, 45</adres>

<tel>0634125412</tel>

</el>

<el

nameOfReader="Авис"

>

<adres>пер. Ленивый, 1</adres>

<tel>0665588687</tel>

</el>

<el

nameOfReader="Симоненко"

>

<adres>ул. Локка, 21</adres>

<tel>0984578211</tel>

</el>

</Reader>

</Data>

<!DOCTYPE html>

<html>

<head>

<title>lab13\_4</title>

<script type="text/javascript">

class Book {

#avtor;

#nameOfBook;

#year;

#price;

constructor(avtor, nameOfBook, year, price) {

this.avtor=avtor;

this.nameOfBook=nameOfBook;

this.year=year;

this.price=price;

}

toString() {

return `Avtor: ${this.avtor}; Book: ${this.nameOfBook}; Year: ${this.year}; Price: ${this.price}; `;

}

set setAvtor(value) {

this.avtor=value;

}

set setNameOfBook(value) {

this.nameOfBook=value;

}

set setYear(value) {

this.year=value;

}

set setPrice(value) {

this.price=value;

}

get getAvtor() {

return this.avtor;

}

get getNameOfBook() {

return this.nameOfBook;

}

get getYear() {

return this.year;

}

get getPrice() {

return this.price;

}

}

class Reader {

#nameOfReader;

#adres;

#tel;

constructor(nameOfReader, adres, tel) {

this.nameOfReader=nameOfReader;

this.adres=adres;

this.tel=tel;

}

toString() {

return `Reader: ${this.nameOfReader}; Adres: ${this.adres}; Tel.: ${this.tel}; `;

}

set setNameOfReader(value) {

this.nameOfReader=value;

}

set setAdres(value) {

this.adres=value;

}

set setTel(value) {

this.tel=value;

}

get getNameOfReader() {

return this.nameOfReader;

}

get getAdres() {

return this.adres;

}

get getTel() {

return this.tel;

}

}

class Note {

#reader;

#book;

#angabeDatum;

#zuruckgabeDatum;

#faktZD;

constructor(Reader, Book, angabeDatum, zuruckgabeDatum, faktZD) {

this.reader=Reader;

this.book=Book;

this.angabeDatum=angabeDatum;

this.zuruckgabeDatum=zuruckgabeDatum;

this.faktZD=faktZD;

}

toString() {

return `${this.reader.get} ${this.book.toString()} Angabedatum: ${this.angabeDatum}; Zurückgabedatum: ${this.zuruckgabeDatum}; Tatsächliches Zurückgabedatum: ${this.faktZD};`;

}

get getReader() {

return this.reader;

}

get getBook() {

return this.book;

}

get getAngabeDatum() {

return this.angabeDatum;

}

get getZuruckgabeDatum() {

return this.zuruckgabeDatum;

}

get getFaktZD() {

return this.faktZD;

}

}

function randomDate(start, end) {

var date = new Date(+start + Math.random() \* (end - start));

day=date.getDay();

month=date.getMonth();

year=date.getFullYear();

var newDate=`${year}.${month}.${day}`;

return newDate;

}

datas=[randomDate(new Date(2012, 0, 1), new Date()), randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date())];

function createTableBook(a, n,y,p) {

var tbody=document.getElementById('tableBook').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=a;

var td2=document.createElement('TD');

td2.innerHTML=n;

var td3=document.createElement('TD');

td3.innerHTML=y;

var td4=document.createElement('TD');

td4.innerHTML=p;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

row.appendChild(td4);

tbody.appendChild(row);

}

function createTableReader(n,a,t) {

var tbody=document.getElementById('tableReader').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=n;

var td2=document.createElement('TD');

td2.innerHTML=a;

var td3=document.createElement('TD');

td3.innerHTML=t;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

tbody.appendChild(row);

}

function createTableNote(r,b,a,z,f) {

var tbody=document.getElementById('tableNote').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=r;

var td2=document.createElement('TD');

td2.innerHTML=b;

var td3=document.createElement('TD');

td3.innerHTML=a;

var td4=document.createElement('TD');

td4.innerHTML=z;

var td5=document.createElement('TD');

td5.innerHTML=f;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

row.appendChild(td4);

row.appendChild(td5);

tbody.appendChild(row);

}

function createFormularBook(a,n,y,p) {

let forma=document.getElementsByClassName('formularBook')[0];

let bookk = document.createElement('fieldset');

bookk.appendChild(document.createElement('legend'));

bookk.getElementsByTagName("legend")[0].innerHTML='Buch';//innerHTML

bookk.appendChild(document.createElement('div'));

bookk.getElementsByTagName('div')[0].innerHTML=`Avtor: ${a};<br> Book: ${n};<br> Year: ${y};<br> Price: ${p};<br>`;

forma.appendChild(bookk);

}

function createFormularReader(n,a,t) {

let forma=document.getElementsByClassName('formularReader')[0];

let readerr=document.createElement('fieldset');

readerr.appendChild(document.createElement('legend'));

readerr.getElementsByTagName('legend')[0].innerHTML='Leser';

readerr.appendChild(document.createElement('div'));

readerr.getElementsByTagName('div')[0].innerHTML=`Reader: ${n};<br> Adres: ${a};<br> Tel.: ${t};<br>`;

forma.appendChild(readerr);

}

function createFormularNote(r,b,a,z,f) {

let forma=document.getElementsByClassName('formularNote')[0];

let notee=document.createElement('fieldset');

notee.appendChild(document.createElement('legend'));

notee.getElementsByTagName('legend')[0].innerHTML='Datenspeicherung';

notee.appendChild(document.createElement('div'));

notee.getElementsByTagName('div')[0].innerHTML=`${r}<br> ${b}<br> Angabedatum: ${a};<br> Zurückgabedatum: ${z};<br> Tatsächliches Zurückgabedatum: ${f};<br>`;

forma.appendChild(notee);

}

</script>

<style type="text/css">

.disabled {

display: none;

}

.abled {

display: block;

}

tr, td, th {

border: 3px inset #cccc00;

text-align: center;

}

tr {

background-color: yellow;

}

table {

margin-right: 15%;

margin-left: 15%;

}

th {

background-color: #e6e600;

}

body {

color: #808000;

font-size: 20px;

}

legend {

background-color: #ffffcc;

border: 1px outset #666600;

width: 30%;

}

fieldset {

border: 1px outset #666600;

background-color: #e6e600;

box-shadow: 0px 0px 5px 5px #cccc00;

}

select, option {

background-color: #ffffcc;

color: #808000;

font-size: 20px;

}

</style>

</head>

<body bgcolor="#ffff66">

<form>

<input type="radio" name="pack" id="radioTable" onclick="checkTable()"><label for="radioTable">Tabele</label><br>

<input type="radio" name="pack" id="radioFormular" onclick="checkTable()"><label for="radioFormular">Formular</label><br>

<select id="list" onchange="checkTable()">

<option disabled selected value="Options">Varianten</option>

<option id="listBook">Buch</option>

<option id="listReader">Leser</option>

<option id="listNote">Datenspeicherung</option>

</select>

<br>

</form>

<br>

<table id="tableBook" width="100%" class="disabled">

<tr><th>Avtor</th><th>Name of Book</th><th>Year</th><th>Price</th></tr>

</table>

<table id="tableReader" width="100%" class="disabled">

<tr><th>Name od Reader</th><th>Adres</th><th>Tel.</th></tr>

</table>

<table id="tableNote" width="100%" class="disabled">

<tr><th>Reader</th><th>Book</th><th>Angabedatum</th><th>Zuruck..datum</th><th>FaktZD</th></tr>

</table>

<form class="formularBook disabled" id="formBuch"></form>

<form class="formularReader disabled" id="formLeser"></form>

<form class="formularNote disabled" id="formDaten"></form>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script>

let arr1 = [];

let arr2=[];

let books = [];

let readers = [];

let notes=[];

var xhttp, textInf;

xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (xhttp.readyState == 4 && xhttp.status == 200) {

textInf = xhttp.responseXML;

arr1 = textInf.getElementsByTagName("Book")[0].children;

arr2 = textInf.getElementsByTagName("Reader")[0].children;

for(let i=0; i < arr1.length; i++){

let avtor = arr1[i].getAttribute('avtor');

let nameOfBook = arr1[i].getAttribute('nameOfBook');

let year = parseInt(arr1[i].children[0].innerHTML);

let price=parseInt(arr1[i].children[1].innerHTML);

books.push(new Book(avtor, nameOfBook, year, price));

}

for(let i=0; i < arr2.length; i++){

let nameOfReader = arr2[i].getAttribute('nameOfReader');

let adres = arr2[i].children[0].innerHTML;

let tel =arr2[i].children[1].innerHTML;

readers.push(new Reader(nameOfReader, adres, tel));

}

books.forEach(book=>{

console.log(book.toString())

});

readers.forEach(reader=>{

console.log(reader.toString())

});

notes=[new Note(books[0], readers[0],datas[0],datas[1],datas[2]), new Note (books[1], readers[1], datas[3], datas[4], datas[5]), new Note (books[2], readers[2], datas[6], datas[7], datas[8]),new Note (books[3], readers[3], datas[9], datas[10], datas[11]) ];

notes.forEach(note=>{

console.log(note.toString())

});

for(let b of books) {

createTableBook(b.getAvtor, b.getNameOfBook, b.getYear, b.getPrice);

}

for(let r of readers) {

createTableReader(r.getNameOfReader, r.getAdres, r.getTel);

}

for(let n of notes) {

createTableNote(n.getReader, n.getBook, n.getAngabeDatum, n.getZuruckgabeDatum, n.getFaktZD);

}

//formulars

for(let b of books) {

createFormularBook(b.getAvtor, b.getNameOfBook, b.getYear, b.getPrice);

}

for(let r of readers) {

createFormularReader(r.getNameOfReader, r.getAdres, r.getTel);

}

for(let n of notes) {

createFormularNote(n.getReader, n.getBook, n.getAngabeDatum, n.getZuruckgabeDatum, n.getFaktZD);

}

}

};

xhttp.open("GET", "http://localhost:8080");

xhttp.send();

//tables

function checkTable() {

let buch=document.getElementById('listBook');

let leser=document.getElementById('listReader');

let daten=document.getElementById('listNote');

if(radioTable.checked){

formBuch.className='formularBook disabled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote disabled';

if(buch.selected){

tableBook.className='abled';

tableReader.className='disabled';

tableNote.className='disabled';

console.log("buchTabele");

}

else if(leser.selected){

tableReader.className='abled';

tableBook.className='disabled';

tableNote.className='disabled';

console.log("leserTabele checked");

} else if(daten.selected){

tableReader.className='disabled';

tableBook.className='disabled';

tableNote.className='abled';

console.log("datenTabele checked");

}

} else if(radioFormular.checked) {

tableReader.className='disabled';

tableBook.className='disabled';

tableNote.className='disabled';

if(buch.selected) {

console.log('radioBuch checked');

formBuch.className='formularBook abled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote disabled';

} else if(leser.selected) {

formBuch.className='formularBook disabled';

formLeser.className='formularReader abled';

formDaten.className='formularNote disabled';

} else if(daten.selected) {

formBuch.className='formularBook disabled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote abled';

}

}

}

</script>

</body>

</html>

Завдання 5(завантаження з JSON (Java Script Object Notation) - файлу)

Модифікувати друге завдання роботи №10 таким чином, щоб дані для масивів-таблиць завантажувалися з JSON файлів та перетворювалися («парсилися») до відповідних об’єктів.

Програмний код:

{ "Book":{

"el":[

{"avtor":"Гельвеций",

"nameOfBook":"Монахиня",

"year":"1678",

"price":"230"},

{ "avtor":"Дидро",

"nameOfBook":"Племянник Рамо",

"year":"1699",

"price":"600"},

{"avtor":"Митчелл",

"nameOfBook":"Унесенные ветром",

"year":"1936",

"price":"789"

},

{"avtor":"Ахматова",

"nameOfBook":"Избранное",

"year":"1993",

"price":"505"

},

{ "avtor":"Прутков",

"nameOfBook":"Полное собрание сочинений",

"year":"1965",

"price":"400"},

{"avtor":"Лондон",

"nameOfBook":"Белый клык",

"year":"1997",

"price":"900"},

{"avtor":"Ситель",

"nameOfBook":"Соло для позвоночника",

"year":"2010",

"price":"221"},

{ "avtor":"Бутромеев",

"nameOfBook":"Древний мир",

"year":"2000",

"price":"333"},

{"avtor":"Гирченко",

"nameOfBook":"Законы Мэрфи",

"year":"2006",

"price":"254"}

]

},

"Reader":{

"el":[

{ "nameOfReader":"Иванов",

"adres":"ул. Тoкарева, 1",

"tel":"0506327533"},

{"nameOfReader":"Петров",

"adres":"ул. Аиста, 13",

"tel":"0506631335"},

{ "nameOfReader":"Сидоров",

"adres":"пер. Советский, 65",

"tel":"0501839016"},

{"nameOfReader":"Конкин",

"adres":"ул. Пушкинская, 89",

"tel":"0663132344"},

{ "nameOfReader":"Либерман",

"adres":"ул. Тюринцева, 5",

"tel":"0985632478"},

{"nameOfReader":"Трас",

"adres":"ул. Миргородская, 6",

"tel":"0503698521"},

{"nameOfReader":"Мюллер",

"adres":"пр. Победы, 45",

"tel":"0634125412"},

{"nameOfReader":"Авис",

"adres":"пер. Ленивый, 1",

"tel":"0665588687"},

{ "nameOfReader":"Симоненко",

"adres":"ул. Локка, 21",

"tel":"0984578211"}

]

}

}

<!DOCTYPE html>

<html>

<head>

<title>lab13\_5</title>

<script type="text/javascript">

class Book {

#avtor;

#nameOfBook;

#year;

#price;

constructor(avtor, nameOfBook, year, price) {

this.avtor=avtor;

this.nameOfBook=nameOfBook;

this.year=year;

this.price=price;

}

toString() {

return `Avtor: ${this.avtor}; Book: ${this.nameOfBook}; Year: ${this.year}; Price: ${this.price}; `;

}

set setAvtor(value) {

this.avtor=value;

}

set setNameOfBook(value) {

this.nameOfBook=value;

}

set setYear(value) {

this.year=value;

}

set setPrice(value) {

this.price=value;

}

get getAvtor() {

return this.avtor;

}

get getNameOfBook() {

return this.nameOfBook;

}

get getYear() {

return this.year;

}

get getPrice() {

return this.price;

}

}

class Reader {

#nameOfReader;

#adres;

#tel;

constructor(nameOfReader, adres, tel) {

this.nameOfReader=nameOfReader;

this.adres=adres;

this.tel=tel;

}

toString() {

return `Reader: ${this.nameOfReader}; Adres: ${this.adres}; Tel.: ${this.tel}; `;

}

set setNameOfReader(value) {

this.nameOfReader=value;

}

set setAdres(value) {

this.adres=value;

}

set setTel(value) {

this.tel=value;

}

get getNameOfReader() {

return this.nameOfReader;

}

get getAdres() {

return this.adres;

}

get getTel() {

return this.tel;

}

}

class Note {

#reader;

#book;

#angabeDatum;

#zuruckgabeDatum;

#faktZD;

constructor(Reader, Book, angabeDatum, zuruckgabeDatum, faktZD) {

this.reader=Reader;

this.book=Book;

this.angabeDatum=angabeDatum;

this.zuruckgabeDatum=zuruckgabeDatum;

this.faktZD=faktZD;

}

toString() {

return `${this.reader.get} ${this.book.toString()} Angabedatum: ${this.angabeDatum}; Zurückgabedatum: ${this.zuruckgabeDatum}; Tatsächliches Zurückgabedatum: ${this.faktZD};`;

}

get getReader() {

return this.reader;

}

get getBook() {

return this.book;

}

get getAngabeDatum() {

return this.angabeDatum;

}

get getZuruckgabeDatum() {

return this.zuruckgabeDatum;

}

get getFaktZD() {

return this.faktZD;

}

}

function randomDate(start, end) {

var date = new Date(+start + Math.random() \* (end - start));

day=date.getDay();

month=date.getMonth();

year=date.getFullYear();

var newDate=`${year}.${month}.${day}`;

return newDate;

}

datas=[randomDate(new Date(2012, 0, 1), new Date()), randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date())];

function createTableBook(a, n,y,p) {

var tbody=document.getElementById('tableBook').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=a;

var td2=document.createElement('TD');

td2.innerHTML=n;

var td3=document.createElement('TD');

td3.innerHTML=y;

var td4=document.createElement('TD');

td4.innerHTML=p;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

row.appendChild(td4);

tbody.appendChild(row);

}

function createTableReader(n,a,t) {

var tbody=document.getElementById('tableReader').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=n;

var td2=document.createElement('TD');

td2.innerHTML=a;

var td3=document.createElement('TD');

td3.innerHTML=t;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

tbody.appendChild(row);

}

function createTableNote(r,b,a,z,f) {

var tbody=document.getElementById('tableNote').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=r;

var td2=document.createElement('TD');

td2.innerHTML=b;

var td3=document.createElement('TD');

td3.innerHTML=a;

var td4=document.createElement('TD');

td4.innerHTML=z;

var td5=document.createElement('TD');

td5.innerHTML=f;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

row.appendChild(td4);

row.appendChild(td5);

tbody.appendChild(row);

}

function createFormularBook(a,n,y,p) {

let forma=document.getElementsByClassName('formularBook')[0];

let bookk = document.createElement('fieldset');

bookk.appendChild(document.createElement('legend'));

bookk.getElementsByTagName("legend")[0].innerHTML='Buch';//innerHTML

bookk.appendChild(document.createElement('div'));

bookk.getElementsByTagName('div')[0].innerHTML=`Avtor: ${a};<br> Book: ${n};<br> Year: ${y};<br> Price: ${p};<br>`;

forma.appendChild(bookk);

}

function createFormularReader(n,a,t) {

let forma=document.getElementsByClassName('formularReader')[0];

let readerr=document.createElement('fieldset');

readerr.appendChild(document.createElement('legend'));

readerr.getElementsByTagName('legend')[0].innerHTML='Leser';

readerr.appendChild(document.createElement('div'));

readerr.getElementsByTagName('div')[0].innerHTML=`Reader: ${n};<br> Adres: ${a};<br> Tel.: ${t};<br>`;

forma.appendChild(readerr);

}

function createFormularNote(r,b,a,z,f) {

let forma=document.getElementsByClassName('formularNote')[0];

let notee=document.createElement('fieldset');

notee.appendChild(document.createElement('legend'));

notee.getElementsByTagName('legend')[0].innerHTML='Datenspeicherung';

notee.appendChild(document.createElement('div'));

notee.getElementsByTagName('div')[0].innerHTML=`${r}<br> ${b}<br> Angabedatum: ${a};<br> Zurückgabedatum: ${z};<br> Tatsächliches Zurückgabedatum: ${f};<br>`;

forma.appendChild(notee);

}

</script>

<style type="text/css">

.disabled {

display: none;

}

.abled {

display: block;

}

tr, td, th {

border: 3px inset #cccc00;

text-align: center;

}

tr {

background-color: yellow;

}

table {

margin-right: 15%;

margin-left: 15%;

}

th {

background-color: #e6e600;

}

body {

color: #808000;

font-size: 20px;

}

legend {

background-color: #ffffcc;

border: 1px outset #666600;

width: 30%;

}

fieldset {

border: 1px outset #666600;

background-color: #e6e600;

box-shadow: 0px 0px 5px 5px #cccc00;

}

select, option {

background-color: #ffffcc;

color: #808000;

font-size: 20px;

}

</style>

</head>

<body bgcolor="#ffff66">

<form>

<input type="radio" name="pack" id="radioTable" onclick="checkTable()"><label for="radioTable">Tabele</label><br>

<input type="radio" name="pack" id="radioFormular" onclick="checkTable()"><label for="radioFormular">Formular</label><br>

<select id="list" onchange="checkTable()">

<option disabled selected value="Options">Varianten</option>

<option id="listBook">Buch</option>

<option id="listReader">Leser</option>

<option id="listNote">Datenspeicherung</option>

</select>

<br>

</form>

<br>

<table id="tableBook" width="100%" class="disabled">

<tr><th>Avtor</th><th>Name of Book</th><th>Year</th><th>Price</th></tr>

</table>

<table id="tableReader" width="100%" class="disabled">

<tr><th>Name od Reader</th><th>Adres</th><th>Tel.</th></tr>

</table>

<table id="tableNote" width="100%" class="disabled">

<tr><th>Reader</th><th>Book</th><th>Angabedatum</th><th>Zuruck..datum</th><th>FaktZD</th></tr>

</table>

<form class="formularBook disabled" id="formBuch"></form>

<form class="formularReader disabled" id="formLeser"></form>

<form class="formularNote disabled" id="formDaten"></form>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script>

let arr1 = [];

let arr2=[];

let books = [];

let readers = [];

let notes=[];

var xhttp, textInf;

xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (xhttp.readyState == 4 && xhttp.status == 200) {

textInf = xhttp.responseText;

arr1 = JSON.parse(textInf).Book.el;

arr2 = JSON.parse(textInf).Reader.el;;

for(let i=0; i < arr1.length; i++){

let avtor = arr1[i].avtor;

let nameOfBook = arr1[i].nameOfBook;

let year = parseInt(arr1[i].year);

let price=parseInt(arr1[i].price);

books.push(new Book(avtor, nameOfBook, year, price));

}

for(let i=0; i < arr2.length; i++){

let nameOfReader = arr2[i].nameOfReader;

let adres = arr2[i].adres;

let tel =arr2[i].tel;

readers.push(new Reader(nameOfReader, adres, tel));

}

books.forEach(book=>{

console.log(book.toString())

});

readers.forEach(reader=>{

console.log(reader.toString())

});

notes=[new Note(books[0], readers[0],datas[0],datas[1],datas[2]), new Note (books[1], readers[1], datas[3], datas[4], datas[5]), new Note (books[2], readers[2], datas[6], datas[7], datas[8]),new Note (books[3], readers[3], datas[9], datas[10], datas[11]) ];

notes.forEach(note=>{

console.log(note.toString())

});

for(let b of books) {

createTableBook(b.getAvtor, b.getNameOfBook, b.getYear, b.getPrice);

}

for(let r of readers) {

createTableReader(r.getNameOfReader, r.getAdres, r.getTel);

}

for(let n of notes) {

createTableNote(n.getReader, n.getBook, n.getAngabeDatum, n.getZuruckgabeDatum, n.getFaktZD);

}

//formulars

for(let b of books) {

createFormularBook(b.getAvtor, b.getNameOfBook, b.getYear, b.getPrice);

}

for(let r of readers) {

createFormularReader(r.getNameOfReader, r.getAdres, r.getTel);

}

for(let n of notes) {

createFormularNote(n.getReader, n.getBook, n.getAngabeDatum, n.getZuruckgabeDatum, n.getFaktZD);

}

}

};

xhttp.open("GET", "http://localhost:8080");

xhttp.send();

//tables

function checkTable() {

let buch=document.getElementById('listBook');

let leser=document.getElementById('listReader');

let daten=document.getElementById('listNote');

if(radioTable.checked){

formBuch.className='formularBook disabled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote disabled';

if(buch.selected){

tableBook.className='abled';

tableReader.className='disabled';

tableNote.className='disabled';

console.log("buchTabele");

}

else if(leser.selected){

tableReader.className='abled';

tableBook.className='disabled';

tableNote.className='disabled';

console.log("leserTabele checked");

} else if(daten.selected){

tableReader.className='disabled';

tableBook.className='disabled';

tableNote.className='abled';

console.log("datenTabele checked");

}

} else if(radioFormular.checked) {

tableReader.className='disabled';

tableBook.className='disabled';

tableNote.className='disabled';

if(buch.selected) {

console.log('radioBuch checked');

formBuch.className='formularBook abled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote disabled';

} else if(leser.selected) {

formBuch.className='formularBook disabled';

formLeser.className='formularReader abled';

formDaten.className='formularNote disabled';

} else if(daten.selected) {

formBuch.className='formularBook disabled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote abled';

}

}

}

</script>

</body>

</html>

Завдання 6

Модифікувати одно з минулих завдань згідно до свого варіанту переписавши завантаження з «колбеку» не «проміс»

Програмний код:

<!DOCTYPE html>

<html>

<head>

<title>lab13\_6</title>

<script type="text/javascript">

class Book {

#avtor;

#nameOfBook;

#year;

#price;

constructor(avtor, nameOfBook, year, price) {

this.avtor=avtor;

this.nameOfBook=nameOfBook;

this.year=year;

this.price=price;

}

toString() {

return `Avtor: ${this.avtor}; Book: ${this.nameOfBook}; Year: ${this.year}; Price: ${this.price}; `;

}

set setAvtor(value) {

this.avtor=value;

}

set setNameOfBook(value) {

this.nameOfBook=value;

}

set setYear(value) {

this.year=value;

}

set setPrice(value) {

this.price=value;

}

get getAvtor() {

return this.avtor;

}

get getNameOfBook() {

return this.nameOfBook;

}

get getYear() {

return this.year;

}

get getPrice() {

return this.price;

}

}

class Reader {

#nameOfReader;

#adres;

#tel;

constructor(nameOfReader, adres, tel) {

this.nameOfReader=nameOfReader;

this.adres=adres;

this.tel=tel;

}

toString() {

return `Reader: ${this.nameOfReader}; Adres: ${this.adres}; Tel.: ${this.tel}; `;

}

set setNameOfReader(value) {

this.nameOfReader=value;

}

set setAdres(value) {

this.adres=value;

}

set setTel(value) {

this.tel=value;

}

get getNameOfReader() {

return this.nameOfReader;

}

get getAdres() {

return this.adres;

}

get getTel() {

return this.tel;

}

}

class Note {

#reader;

#book;

#angabeDatum;

#zuruckgabeDatum;

#faktZD;

constructor(Reader, Book, angabeDatum, zuruckgabeDatum, faktZD) {

this.reader=Reader;

this.book=Book;

this.angabeDatum=angabeDatum;

this.zuruckgabeDatum=zuruckgabeDatum;

this.faktZD=faktZD;

}

toString() {

return `${this.reader.get} ${this.book.toString()} Angabedatum: ${this.angabeDatum}; Zurückgabedatum: ${this.zuruckgabeDatum}; Tatsächliches Zurückgabedatum: ${this.faktZD};`;

}

get getReader() {

return this.reader;

}

get getBook() {

return this.book;

}

get getAngabeDatum() {

return this.angabeDatum;

}

get getZuruckgabeDatum() {

return this.zuruckgabeDatum;

}

get getFaktZD() {

return this.faktZD;

}

}

function randomDate(start, end) {

var date = new Date(+start + Math.random() \* (end - start));

day=date.getDay();

month=date.getMonth();

year=date.getFullYear();

var newDate=`${year}.${month}.${day}`;

return newDate;

}

datas=[randomDate(new Date(2012, 0, 1), new Date()), randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date()),randomDate(new Date(2012, 0, 1), new Date())];

function createTableBook(a, n,y,p) {

var tbody=document.getElementById('tableBook').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=a;

var td2=document.createElement('TD');

td2.innerHTML=n;

var td3=document.createElement('TD');

td3.innerHTML=y;

var td4=document.createElement('TD');

td4.innerHTML=p;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

row.appendChild(td4);

tbody.appendChild(row);

}

function createTableReader(n,a,t) {

var tbody=document.getElementById('tableReader').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=n;

var td2=document.createElement('TD');

td2.innerHTML=a;

var td3=document.createElement('TD');

td3.innerHTML=t;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

tbody.appendChild(row);

}

function createTableNote(r,b,a,z,f) {

var tbody=document.getElementById('tableNote').getElementsByTagName('TBODY')[0];

var row=document.createElement('TR');

var td1=document.createElement('TD');

td1.innerHTML=r;

var td2=document.createElement('TD');

td2.innerHTML=b;

var td3=document.createElement('TD');

td3.innerHTML=a;

var td4=document.createElement('TD');

td4.innerHTML=z;

var td5=document.createElement('TD');

td5.innerHTML=f;

row.appendChild(td1);

row.appendChild(td2);

row.appendChild(td3);

row.appendChild(td4);

row.appendChild(td5);

tbody.appendChild(row);

}

function createFormularBook(a,n,y,p) {

let forma=document.getElementsByClassName('formularBook')[0];

let bookk = document.createElement('fieldset');

bookk.appendChild(document.createElement('legend'));

bookk.getElementsByTagName("legend")[0].innerHTML='Buch';//innerHTML

bookk.appendChild(document.createElement('div'));

bookk.getElementsByTagName('div')[0].innerHTML=`Avtor: ${a};<br> Book: ${n};<br> Year: ${y};<br> Price: ${p};<br>`;

forma.appendChild(bookk);

}

function createFormularReader(n,a,t) {

let forma=document.getElementsByClassName('formularReader')[0];

let readerr=document.createElement('fieldset');

readerr.appendChild(document.createElement('legend'));

readerr.getElementsByTagName('legend')[0].innerHTML='Leser';

readerr.appendChild(document.createElement('div'));

readerr.getElementsByTagName('div')[0].innerHTML=`Reader: ${n};<br> Adres: ${a};<br> Tel.: ${t};<br>`;

forma.appendChild(readerr);

}

function createFormularNote(r,b,a,z,f) {

let forma=document.getElementsByClassName('formularNote')[0];

let notee=document.createElement('fieldset');

notee.appendChild(document.createElement('legend'));

notee.getElementsByTagName('legend')[0].innerHTML='Datenspeicherung';

notee.appendChild(document.createElement('div'));

notee.getElementsByTagName('div')[0].innerHTML=`${r}<br> ${b}<br> Angabedatum: ${a};<br> Zurückgabedatum: ${z};<br> Tatsächliches Zurückgabedatum: ${f};<br>`;

forma.appendChild(notee);

}

</script>

<style type="text/css">

.disabled {

display: none;

}

.abled {

display: block;

}

tr, td, th {

border: 3px inset #cccc00;

text-align: center;

}

tr {

background-color: yellow;

}

table {

margin-right: 15%;

margin-left: 15%;

}

th {

background-color: #e6e600;

}

body {

color: #808000;

font-size: 20px;

}

legend {

background-color: #ffffcc;

border: 1px outset #666600;

width: 30%;

}

fieldset {

border: 1px outset #666600;

background-color: #e6e600;

box-shadow: 0px 0px 5px 5px #cccc00;

}

select, option {

background-color: #ffffcc;

color: #808000;

font-size: 20px;

}

</style>

</head>

<body bgcolor="#ffff66">

<form>

<input type="radio" name="pack" id="radioTable" onclick="checkTable()"><label for="radioTable">Tabele</label><br>

<input type="radio" name="pack" id="radioFormular" onclick="checkTable()"><label for="radioFormular">Formular</label><br>

<select id="list" onchange="checkTable()">

<option disabled selected value="Options">Varianten</option>

<option id="listBook">Buch</option>

<option id="listReader">Leser</option>

<option id="listNote">Datenspeicherung</option>

</select>

<br>

</form>

<br>

<table id="tableBook" width="100%" class="disabled">

<tr><th>Avtor</th><th>Name of Book</th><th>Year</th><th>Price</th></tr>

</table>

<table id="tableReader" width="100%" class="disabled">

<tr><th>Name od Reader</th><th>Adres</th><th>Tel.</th></tr>

</table>

<table id="tableNote" width="100%" class="disabled">

<tr><th>Reader</th><th>Book</th><th>Angabedatum</th><th>Zuruck..datum</th><th>FaktZD</th></tr>

</table>

<form class="formularBook disabled" id="formBuch"></form>

<form class="formularReader disabled" id="formLeser"></form>

<form class="formularNote disabled" id="formDaten"></form>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script>

let arr1 = [];

let arr2=[];

let books = [];

let readers = [];

let notes=[];

var xhttp, textInf;

function get(url) {

return new Promise(function(succeed, fail) {

var request = new XMLHttpRequest();

request.open("GET", url, true);

request.addEventListener("load", function() {

if (request.status < 400)

succeed(request.responseText);

else

fail(new Error("Request failed: " + request.statusText));

});

request.addEventListener("error", function() {

fail(new Error("Network error"));

});

request.send();

});

}

get("http://localhost:8080").then(function(response) {

return response;

}).then(function(data) {

let arrays = data.split('||');

arr1 = arrays[0].split('#');

arr2 = arrays[1].split('#');

for(let i=0; i < arr1.length; i++){

let obj = arr1[i].split(';');

books.push(new Book(obj[0],obj[1],parseInt(obj[2]), parseInt(obj[3])));

}

for(let i=0; i < arr2.length; i++){

let obj = arr2[i].split(';');

readers.push(new Reader(obj[0],obj[1],obj[2]));

}

books.forEach(book=>{

console.log(book.toString())

});

readers.forEach(reader=>{

console.log(reader.toString())

});

notes=[new Note(books[0], readers[0],datas[0],datas[1],datas[2]), new Note (books[1], readers[1], datas[3], datas[4], datas[5]), new Note (books[2], readers[2], datas[6], datas[7], datas[8]),new Note (books[3], readers[3], datas[9], datas[10], datas[11]) ];

notes.forEach(note=>{

console.log(note.toString())

});

for(let b of books) {

createTableBook(b.getAvtor, b.getNameOfBook, b.getYear, b.getPrice);

}

for(let r of readers) {

createTableReader(r.getNameOfReader, r.getAdres, r.getTel);

}

for(let n of notes) {

createTableNote(n.getReader, n.getBook, n.getAngabeDatum, n.getZuruckgabeDatum, n.getFaktZD);

}

//formulars

for(let b of books) {

createFormularBook(b.getAvtor, b.getNameOfBook, b.getYear, b.getPrice);

}

for(let r of readers) {

createFormularReader(r.getNameOfReader, r.getAdres, r.getTel);

}

for(let n of notes) {

createFormularNote(n.getReader, n.getBook, n.getAngabeDatum, n.getZuruckgabeDatum, n.getFaktZD);

}

books.forEach(book=>{

console.log(book.toString())

});

readers.forEach(reader=>{

console.log(reader.toString())

});

});

//tables

function checkTable() {

let buch=document.getElementById('listBook');

let leser=document.getElementById('listReader');

let daten=document.getElementById('listNote');

if(radioTable.checked){

formBuch.className='formularBook disabled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote disabled';

if(buch.selected){

tableBook.className='abled';

tableReader.className='disabled';

tableNote.className='disabled';

console.log("buchTabele");

}

else if(leser.selected){

tableReader.className='abled';

tableBook.className='disabled';

tableNote.className='disabled';

console.log("leserTabele checked");

} else if(daten.selected){

tableReader.className='disabled';

tableBook.className='disabled';

tableNote.className='abled';

console.log("datenTabele checked");

}

} else if(radioFormular.checked) {

tableReader.className='disabled';

tableBook.className='disabled';

tableNote.className='disabled';

if(buch.selected) {

console.log('radioBuch checked');

formBuch.className='formularBook abled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote disabled';

} else if(leser.selected) {

formBuch.className='formularBook disabled';

formLeser.className='formularReader abled';

formDaten.className='formularNote disabled';

} else if(daten.selected) {

formBuch.className='formularBook disabled';

formLeser.className='formularReader disabled';

formDaten.className='formularNote abled';

}

}

}

</script>

</body>

</html>