|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

**Отчет по лабораторной работе № 2 по курсу**

**«Информационные системы и технологии»**

" Парсинг XML с использованием JavaScript"

Вариант № <6>

Кобзева Елизавета

СГН3-61Б

2025 г.

**Задание**

Создать HTML страницу с встроенным скриптом, который будет обрабатывать выданный вам XML файл. Вам необходимо вывести на экран все элементы третьего, последнего уровня вложенности одного из элементов второго уровня.

**Дополнительное задание**

В HTML файле создать поле ввода, в которое вводиться номер элемента второго уровня, для которого выведутся элементы третьего уровня.

Порядок выполнения работы

**Код файла xml**

<root>

<catalog>

<head>Зарубнжная литература</head>

<book>

<title>Дюна</title>

<author>Фрэнк Герберт</author>

<year>1965</year>

<janre>Научная фантастика</janre>

</book>

<book>

<title>Нейромант</title>

<author>Уильям Гибсон</author>

<year>1984</year>

<janre>Киберпанк</janre>

</book>

<book>

<title>Шурлок Холмс</title>

<author>Артур Конан-Дойл</author>

<year>1870</year>

<janre>Детектив</janre>

</book>

<book>

<title>Три товарища</title>

<author>Эрих Ремарк</author>

<year>1935</year>

<janre>Драма</janre>

</book>

</catalog>

<catalog>

<head>Русская литература</head>

<book>

<title>Преступление и наказание</title>

<author>Фёдор Достоевский</author>

<year>1866</year>

<janre>Роман</janre>

</book>

<book>

<title>Война и мир</title>

<author>Лев Толстой</author>

<year>1869</year>

<janre>Исторический роман</janre>

</book>

<book>

<title>Евгений Онегин</title>

<author>Александр Пушкин</author>

<year>1830</year>

<janre>Роман в стихах</janre>

</book>

<book>

<title>Мастер и Маргарита</title>

<author>Михаил Булгаков</author>

<year>1967</year>

<janre>Мистика</janre>

</book>

</catalog>

</root>

**Код файла Css**

.input-container {

margin: 20px auto;

text-align: center;

}

.input-container input {

padding: 10px;

font-size: 16px;

border: 1px solid #ddd;

border-radius: 5px;

width: 200px;

}

.catalog-container {

font-family: Arial, sans-serif;

padding: 20px;

background-color: #f9f9f9;

border: 1px solid #ddd;

border-radius: 8px;

max-width: 1200px;

margin: 20px auto; /\* Отступ между каталогами \*/

}

.catalog-head {

font-size: 24px;

font-weight: bold;

color: #333;

margin-bottom: 20px;

text-align: center;

}

.books-container {

display: flex;

flex-wrap: wrap;

gap: 16px;

justify-content: flex-start;

}

.book {

background-color: #fff;

border: 1px solid #ddd;

border-radius: 8px;

padding: 16px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

width: calc(25% - 16px);

box-sizing: border-box;

transition: background-color 0.3s ease;

}

.book.selected {

background-color: #e0f7fa; /\* Подсветка выбранной книги \*/

border: 2px solid #007bff;

}

.book-title {

font-size: 18px;

font-weight: bold;

color: #007bff;

margin-bottom: 8px;

}

.book-author {

font-size: 16px;

color: #555;

margin-bottom: 8px;

}

.book-year {

font-size: 14px;

color: #777;

margin-bottom: 8px;

}

.book-janre {

font-size: 14px;

color: #28a745;

font-style: italic;

}

**Код файла JS**

import React, { useState } from 'react';

import './App.css';

function App() {

const [firstCatalog, setFirstCatalog] = useState([]); // Первый каталог

const [secondCatalog, setSecondCatalog] = useState([]); // Второй каталог

const [selectedBookIndex, setSelectedBookIndex] = useState(null); // Выбранная книга

// Загрузка XML с помощью XMLHttpRequest

const loadXML = () => {

const xhr = new XMLHttpRequest();

xhr.open('GET', '/data.xml', true); // Асинхронный запрос

xhr.onload = () => {

if (xhr.status === 200) {

const parser = new DOMParser();

const xmlDoc = parser.parseFromString(xhr.responseText, 'text/xml');

// Извлечение первого каталога

const firstCatalogData = xmlDoc.getElementsByTagName('catalog')[0];

const firstHead = firstCatalogData.getElementsByTagName('head')[0].textContent;

const firstBooks = firstCatalogData.getElementsByTagName('book');

const firstCatalogBooks = [];

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

const title = firstBooks[i].getElementsByTagName('title')[0].textContent;

const author = firstBooks[i].getElementsByTagName('author')[0].textContent;

const year = firstBooks[i].getElementsByTagName('year')[0].textContent;

const janre = firstBooks[i].getElementsByTagName('janre')[0].textContent;

firstCatalogBooks.push({ title, author, year, janre });

}

setFirstCatalog({ head: firstHead, books: firstCatalogBooks });

// Извлечение второго каталога

const secondCatalogData = xmlDoc.getElementsByTagName('catalog')[1];

const secondHead = secondCatalogData.getElementsByTagName('head')[0].textContent;

const secondBooks = secondCatalogData.getElementsByTagName('book');

const secondCatalogBooks = [];

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

const title = secondBooks[i].getElementsByTagName('title')[0].textContent;

const author = secondBooks[i].getElementsByTagName('author')[0].textContent;

const year = secondBooks[i].getElementsByTagName('year')[0].textContent;

const janre = secondBooks[i].getElementsByTagName('janre')[0].textContent;

secondCatalogBooks.push({ title, author, year, janre });

}

setSecondCatalog({ head: secondHead, books: secondCatalogBooks });

} else {

console.error('Ошибка при загрузке XML:', xhr.statusText);

}

};

xhr.onerror = () => {

console.error('Ошибка при выполнении запроса.');

};

xhr.send();

};

// Загружаем XML при монтировании компонента

React.useEffect(() => {

loadXML();

}, []);

// Обработчик изменения поля ввода

const handleInputChange = (event) => {

const index = parseInt(event.target.value, 10);

if (!isNaN(index) && index >= 0) {

setSelectedBookIndex(index);

} else {

setSelectedBookIndex(null);

}

};

return (

<div>

{/\* Поле ввода для выбора книги \*/}

<div className="input-container">

<input

type="number"

placeholder="Введите номер книги"

onChange={handleInputChange}

min="0"

/>

</div>

{/\* Первый каталог \*/}

{firstCatalog.head && (

<div className="catalog-container">

<div className="catalog-head">{firstCatalog.head}</div>

<div className="books-container">

{firstCatalog.books.map((book, index) => (

<div

key={index}

className={`book ${selectedBookIndex === index ? 'selected' : ''}`}

>

<div className="book-title">{book.title}</div>

<div className="book-author">{book.author}</div>

<div className="book-year">{book.year}</div>

<div className="book-janre">{book.janre}</div>

</div>

))}

</div>

</div>

)}

{/\* Второй каталог \*/}

{secondCatalog.head && (

<div className="catalog-container">

<div className="catalog-head">{secondCatalog.head}</div>

<div className="books-container">

{secondCatalog.books.map((book, index) => {

// Учитываем смещение индекса для второго каталога

const globalIndex = firstCatalog.books ? firstCatalog.books.length + index : index;

return (

<div

key={globalIndex}

className={`book ${selectedBookIndex === globalIndex ? 'selected' : ''}`}

>

<div className="book-title">{book.title}</div>

<div className="book-author">{book.author}</div>

<div className="book-year">{book.year}</div>

<div className="book-janre">{book.janre}</div>

</div>

);

})}

</div>

</div>

)}

</div>

);

}

export default App;

**Скриншоты работающей программы**



