

УДК 004.91:004.422

Смоктій К.В., к.е.н., доцент,  
доцент кафедри комп'ютерних  
наук та інформаційних технологій  
Чайковський П.А., студентка 2 курсу  
спеціальності 125 «Кибербезпека»

## ОСОБЛИВОСТІ ВІДОБРАЖЕННЯ (РЕНДЕРИНГУ) ІЄРАРХІЧНИХ СТРУКТУР В WEB-СТОРІНКАХ

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Сучасний розвиток інформаційних технологій характеризується високим рівнем використання мережі інтернет та інтернет-додатків, великою різноманітністю інструментальних засобів для їх реалізації. Проаналізуємо зручність та складність відображення (рендерингу) ієрархічних структур в web-сторінках на прикладі інформації про організаційну структуру, що наведено на рис. 1.

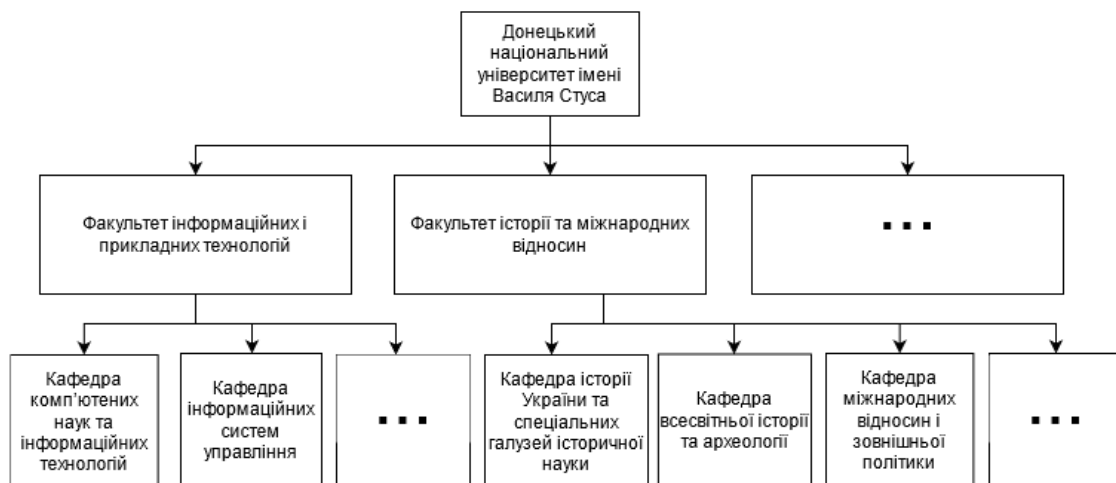


Рисунок 1 – Приклад організаційної структури ДонНУ

Для відображення (рендерингу) ієрархічних структур (деревоподібної структури) в web-сторінках будемо використовувати DOM-модель (document object model) [1] або об'єктна модель документу, яка представляє ієрархію HTML елементів та може бути інтерпретована як граф, де вершини - елементи, у яких можуть бути наслідники [1]. Використаємо для відображення ієрархічних структур безпосередньо мову розмітки гіпертексту HTML (HyperText Markup Language), скриптову мову програмування JavaScript (JS) [2], а також відкриту JavaScript бібліотеку для створення інтерфейсів користувача React [3]. Проаналізуємо отриманий результат.



```

const OptionsList = ({ options, selectedOptions, onChange, nestLevel = 0 }) => {

  const handleCheckboxClicked = (selectedOptionId) => {
    // is currently selected
    if (selectedOptions[selectedOptionId]) {
      // remove selected key from options list
      delete selectedOptions[selectedOptionId];
    } else {
      // is not currently selected
      // Add selected key to optionsList
      selectedOptions[selectedOptionId] = {};
    }
    // call onChange function given by parent
    onChange(selectedOptions);
  };

  const handleSubOptionsListChange = (optionId, subSelections) => {
    // add sub selections to current optionId
    selectedOptions[optionId] = subSelections;
    // call onChange function given by parent
    onChange(selectedOptions);
  };

```

Рисунок 4 – Реалізація рекурсивного компонента OptionList з використання React

```

const renderUser = (arr, node, nestLevel = 0) => {
  return arr.map((x) => {
    const newListItem = renderListItem();
    const option = renderOption(x, nestLevel);
    newListItem.appendChild(option);

    if (x.subOptions.length > 0) {
      const newSubList = renderList();
      newListItem.appendChild(newSubList);
      option.onclick = () => {
        if (isThereChildren(newSubList)) removeChildren(newSubList);
        else renderUser(x.subOptions, newSubList, nestLevel + 1);
      };
    }
    if (node) node.appendChild(newListItem);
    return newListItem;
  });
};

```

Рисунок 5 – Використання JavaScript для відображення ієрархічної структури

Результат використання JavaScript та React наведено на рис. 6 (відображення має ідентичне представлення).

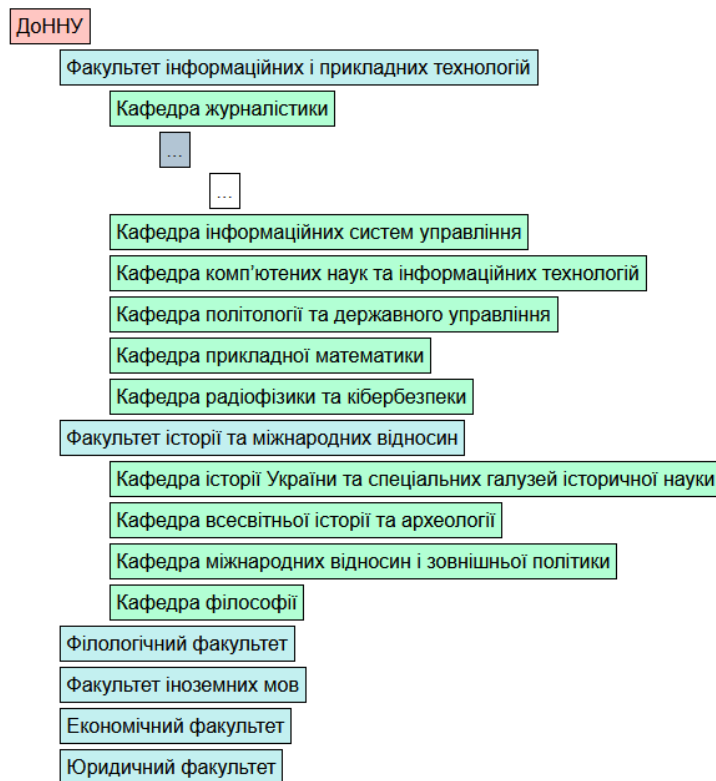


Рисунок 6 – Результат використання JavaScript та React

Зазначимо, що при використанні статичного HTML для відображення ієрархічної структури не дозволяє створювати ефективні односторінкові додатки та потребує обов'язкового повного перезавантаження сторінки для її зміни. При використанні React отримуємо більше коду, ніж при використанні безпосередньо JavaScript, але використання бібліотеки React надає додаткові можливості при масштабуванні та гнучкості при подальшій модифікації [4].

Список використаної літератури:

1. Електронний ресурс. Що таке об'єктна модель документа? Режим доступу: <https://www.w3.org/TR/WD-DOM/introduction.html> (дата звертання: 25.03.2021)
2. Електронний ресурс. JavaScript. Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звертання: 25.03.2021)
3. Електронний ресурс. Рендеринг елементів з використанням React. Режим доступу: <https://ru.reactjs.org/docs/rendering-elements.html> (дата звертання: 25.03.2021)
4. Електронний ресурс. Приклад створених файлів та допоміжні посилання. Режим доступу: [https://github.com/ruxxzebre/Example\\_PIT\\_2021](https://github.com/ruxxzebre/Example_PIT_2021) (дата звертання: 25.03.2021)

**УДК 004.4**

Солодун Т.Р., студентка 2 курсу  
спеціальності 122 «Комп'ютерні науки»  
Ніколюк П.К., д.ф.-м.н., професор,  
професор кафедри комп'ютерних наук  
та інформаційних технологій