МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Факультет радіотехніки, комп’ютерних систем і інфокомунікацій

Кафедра інформаційно-комунікаційних технологій ім. О. О. Зеленського

# Лабораторна робота № 12

з дисципліни: Web-програмирование

(назва дисципліни)

на тему: Стилизация компонентов в React.js

Виконала: студент 3 курсу

групи № 539 напряму підготовки (спеціальності) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(шифр і назва напряму підготовки (спеціальності))

\_\_\_Стенякін А\_\_\_\_\_\_\_\_\_\_

(прізвище й ініціали студента)

Прийняв:Проскура Г.А.

(посада, науковий ступінь, прізвище й ініціали)

Національна шкала: \_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_

Харків – 2020

Лабораторная работа 12

ЦЕЛЬ РАБОТЫ

Научиться применять стилизацию к компонентам React.js,

использовать динамические классы.

ХОД РАБОТЫ

1.Изучите теоретические сведения в лекции 12.

2.Стилизируйте приложение, созданное в лабораторной работе

No11:

1.Создайте в компоненте поле ввода текста input, когда

поле input будет пустое, его граница будет стандартного

цвета, иначе - менять свой цвет.

2.При наведении на отдельный экземпляр компонента

должны изменяться его границы, тень и курсор.

3.Сохраните созданные изменения на гитхаб, отправьте в

оригинальный репозиторий pull request.

Теоретические сведения

Компоненты React — это строительные блоки, из которых мы создаём интерфейсы для приложений. То, как мы стилизуем визуальные элементы, во многом определяет впечатления пользователей о нашем бренде и продукте.

Не существует одного правильного способа стилизации компонентов. Способ стилизации нужно выбирать исходя из конкретного случая и личных предпочтений; но прежде всего, выбор зависит от архитектурных особенностей проекта, например: глобального пространства имён, зависимостей, повторного использования, масштабируемости, устранение «мёртвого кода» и т.д.

React позволяет добавлять CSS непосредственно в код в качестве атрибутов, и передавать его элементам.

В React, inline стили записываются, как объект. Ключ такого объекта является именем стиля; имя объекта записывается в стиле camelCase; значения стиля записываются внутри объекта, обычно строкой.

Для атрибута style, объект JavaScript и его свойства имеют больший приоритет, чем строка CSS. Это согласовано со свойством JavaScript в DOM style, это более эффективно и предотвращает дыры в безопасности XSS.

**CSS в коде JS** для абстракции CSS на уровне компонентов — это использование JavaScript для описания стилей, декларативным и пригодным для дальнейшей поддержки способом. Благодаря релизу проекта [styled-components](https://github.com/styled-components/styled-components" \t "_blank) (автор [Max Stoiber](https://medium.com/@mxstbr" \t "_blank)), этот способ стал более популярен, чем когда либо (далее подробней).

В чём же разница между двумя способами, CSS-in-JS и Inline? [Говоря простым языком](https://hackernoon.com/all-you-need-to-know-about-css-in-js-984a72d48ebc), если вы используете способ CSS-in-JS, то вы прикрепляете тег <style> в DOM, а с Inline стилем, вы прикрепляете свойство к DOM узлу.

В React, способ CSS-in-JS подталкивает вас думать и создавать стили, абстрагируя до уровня компонентов. Этот способ делает акцент на принципах модульности и изоляции, юнит тестирования, DRY (Don’t repeat yourself) и так далее. Это головоломка, но зачастую практичная.

Идея, рождённая в [австралийском баре](https://tylermcginnis.com/podcast/max-stoiber/), выросла до проекта, собравшего 20 000 лайков. Этот проект упрощает использование CSS в React components, определяя styled-components с упакованными стилями, без промежуточного слоя CSS классов.

Чтобы создать Styled-components нужно определить компоненты, используя шаблон буквенной нотации ES6. CSS свойства добавляются в компоненты так же, как вы обычно делаете это с CSS. После парсинга JS, styled-components сгенерирует уникальные имена классов, и внедрит CSS в DOM.

CSS модуль — это CSS файл, в котором все имена классов и анимаций имеют локальную область видимости по умолчанию.

У каждого компонента React есть собственный CSS файл, который ограничен этим компонентом. Чтобы стилизовать компонент React, просто создайте для него CSS-файл, который будет содержать стили.

Во время сборки, имена локальных классов сопоставляются и экспортируются как литерал объекта JS для React. То же происходит и с модифицированной версией входного CSS с переименованными классами. В результате вам не нужно так много возиться с глобальными стилями. При масштабировании проектов у вас будет меньше переопределений и проблем.

**Sass** — это скриптовый язык препроцессора, который интерпретируется или компилируется в CSS. Синтаксически он похож на язык [Haml](https://en.wikipedia.org/wiki/Haml" \t "_blank). Sass использует отступы для разделения блоков кода и символы новой строки для разделения правил.

«**Scss»**(Sassy CSS) имеет более новый синтаксис, в нём такое же форматирование блоков как в CSS. В Scss фигурные скобки используются для обозначения блоков кода, а точки с запятой для разделения строк в блоке. Sass поддерживается в create-react-app 2.0 [по умолчанию](https://reactjs.org/blog/2018/10/01/create-react-app-v2.html). Он становится всё более популярным способом стилизовать компоненты.

Код

MyInfo.js

import React from "react";

import './MyInfo.css'

class App extends Component {

<h2>Известные люди</h2>

state={

people:[

<Name={<h1>'Серджо Маттарелла'</h1> итальянский государственный деятель, юрист, политик, действующий президент Италии c 3 февраля 2015 года.

Член шести составов правительства Италии в 1987—1990 и 1998—2001 годах, судья Конституционного суда Италии (2011—2015)./>

<a href="https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D1%82%D0%B0%D1%80%D0%B5%D0%BB%D0%BB%D0%B0,\_%D0%A1%D0%B5%D1%80%D0%B4%D0%B6%D0%BE" style="text-align: center"><img src="https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D0%BB%D0%B0,\_%D0%9D%D0%B8%D0%BA%D0%BE%D0%BB%D0%B0" alt="Foto"></a>

<Name={<h1>Франк-Вальтер Штайнмайер (нем. Frank-Walter Steinmeier; род. 5 января 1956, Детмольд, ФРГ) — немецкий государственный деятель, политик, действующий Федеральный президент ФРГ с 19 марта 2017 года.

В прошлом — министр иностранных дел Германии (2005—2009 и 2013—2017), вице-канцлер Германии (2007—2009), член Социал-демократической партии Германии с 1975 года. />

<a href="https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D0%BB%D0%B0,\_%D0%9D%D0%B8%D0%BA%D0%BE%D0%BB%D0%B0" class="title-lecture" style="text-align: center"><img src="https://ru.wikipedia.org/wiki/%D0%A8%D1%82%D0%B0%D0%B9%D0%BD%D0%BC%D0%B0%D0%B9%D0%B5%D1%80,\_%D0%A4%D1%80%D0%B0%D0%BD%D0%BA-%D0%92%D0%B0%D0%BB%D1%8C%D1%82%D0%B5%D1%80" alt="Foto"></a>

<Name={<h1>Елизаве́та II (англ. Elizabeth II), полное имя — Елизаве́та Алекса́ндра Мари́я (англ. Elizabeth Alexandra Mary; род. 21 апреля 1926, Мейфэр, Вестминстер, Лондон, Англия, Великобритания) — царствующая королева Великобритании и королевств Содружества из Виндзорской династии. Верховный главнокомандующий вооруженными силами Великобритании. Верховный правитель Церкви Англии. Глава Содружества Наций. Действующий монарх в пятнадцати независимых государствах: Австралии, Антигуа и Барбуде, Багамских Островах, Барбадосе, Белизе, Гренаде, Канаде, Новой Зеландии, Папуа — Новой Гвинее, Сент-Винсенте и Гренадинах, Сент-Китсе и Невисе, Сент-Люсии, Соломоновых Островах, Тувалу, Ямайке. />

<a href="https://ru.wikipedia.org/wiki/%D0%A1%D0%B8\_%D0%A6%D0%B7%D0%B8%D0%BD%D1%8C%D0%BF%D0%B8%D0%BD" class="title-lecture" style="text-align: center"><img src="https://ru.wikipedia.org/wiki/%D0%95%D0%BB%D0%B8%D0%B7%D0%B0%D0%B2%D0%B5%D1%82%D0%B0\_II" alt="Foto"></a>

],

pageTitle :'React components'

}

changeTitleHandler = (newTitle) =>{

this.setState({

pageTitle: newTitle

})

}

onChangeName (name, index){

const people = this.state.people[index]

people.name = name

const people = [...this.state.people]

people[index] = people

this.setState({people})

}

render() {

let people = null

people = this.state.people.map(( people, index) =>{

return(

<People

key={index}

name={people.name}

year={people.year}

specialization={people.specialization}

onChangeTitle={this.changeTitleHandler.bind(this, people.name)}

onChangeName={event => this.onChangeName(event.target.value,

index)}

/>

)

})

return (

<div className="App">

<h2>{this.state.pageTitle}</h2>

<div style={{

paddingTOP: '60px'

}}>

{people}

</div>

</div>

);

}

}

export default App;

<input

type="text"

onChange={props.onChangeName}

value={props.name}

className={inputClasses.join(' ')}

/>

<button onClick={props.onChangeTitle}>Up</button>

</div>

);

}

}

export default App;

App.js

import React from "react";

import'./MyInfo.scss'

import Radium from 'radium'

const MyInfo = props =>{

export default props => (

<div style={{

boxShadow: '8px 8px 8px 0 rgba(0, 0, 0, 0.14)',

borderRadius: '5px',

display: 'inline-block',

width: '30%',

marginRight: '25px',

border: '2px solid #ccc',

height: '512px'

}}>

export default props => {

const inputClasses = ['input']

if(props.name !== ''){

inputClasses.push('green')

}else {

inputClasses.push('red')

}

return(

const style = {

border: '2px solid #ccc',

boxShadow: '0 4px 5px 0 rgba(0, 0, 0, 0.14)'

':hover':{

border: '3px solid #aaa',

boxShadow: '0 4px 15px 0 rgba(0, 0, 0, .25)', cursor: 'pointer'

}

}

<div className='MyInfo' style={style}>

<h1>Name: <strong>{props.name}</strong></h1>

<p>Year born:{props.year}</p>

<p>Specialization: {props.specialization}</p>

<input type="text" onChange={props.onChangeName} value={props.name} className={inputClasses.join(' ')}

/>

<button onClick={props.onChangeTitle}>Click</button>

</div>

)

}

}

export default Radium(MyInfo)

**Вывод**

В даной лабораторной работе было созданно одностраничное стилизированое приложение на основе React. На основе 11 и 12 лекций была переделана начальная веб-страница. И до работана поле ввода меняет свой цвет при вводе данных. Так же при наведении на отдельный экземпляр компонента изменяются границы, тень и курсор.