***Задание 1.*** Экспортируйте компонент. Примените экспорт по умолчанию

function Profile() {

return (

<img

src="https://i.imgur.com/lICfvbD.jpg"

alt="Aklilu Lemma"

/>

);

}

export default Profile

***Задание 2***. Исправьте оператор return (возврата)

export default function Profile() {

return {

<img src="https://i.imgur.com/jA8hHMpm.jpg" alt="Katsuko Saruhashi" />

}

}

***Задание 3.*** Найдите и исправьте ошибку

function Profile() {

return {

<img

src="https://i.imgur.com/QIrZWGIs.jpg"

alt="Alan L. Hart"

/>

}

}

export default function Gallery() {

return {

<section>

<h1>Amazing scientists</h1>

<Profile />

<Profile />

<Profile />

</section>

}

}

***Задание 4.*** Напишите Congratulations компонент, который возвращает текст <h1>Good job!</h1>. Выполните именованный экспорт.

const Congratulation = () => {

return {

<h1>Good job!</h1>

}

export {Congratulations}

***Задание 5.*** Нижеприведенный код выдает ошибку Objects are not valid as a React child… . Исправьте ее.

const Person = {

name: ‘[Брендан Эйх](https://www.google.com/search?sca_esv=569725060&sxsrf=AM9HkKkMd0yvkJhSMLuyylB_jxu6GFW_ZQ:1696087375833&q=%D0%91%D1%80%D0%B5%D0%BD%D0%B4%D0%B0%D0%BD+%D0%AD%D0%B9%D1%85&stick=H4sIAAAAAAAAAONgVuLUz9U3MDE2KrZ8xGjCLfDyxz1hKe1Ja05eY1Tl4grOyC93zSvJLKkUEudig7J4pbi5ELp4FrGKXph4seHC1gt7L2y5sOHCXoULay_svNgKAF18oRZcAAAA&sa=X&ved=2ahUKEwjmrq3s0dKBAxWVExAIHfBOCpkQzIcDKAB6BAgTEAE)’,

theme: {

backgroundColor: ‘black’,

color: ‘pink’

}

};

export default function TodoList({person}) {

return {

<div style={person.theme}>

<h1>{person} – создатель языка программирования JS</h1>

<img

className="avatar"

src="https://i.imgur.com/7vQD0fPs.jpg"

alt="Gregorio Y. Zara"

/>

<ul>

<li key = {person.name}>Improve the videophone</li>

<li key = {person.theme.backgroundColor}>Prepare aeronautics lectures</li>

<li key = {person.theme.color}>Work on the alcohol-fuelled engine</li>

</ul>

</div>

}

}

***Задание 6.*** Переместите URL-адрес изображения в person объект. Используйте соответствующее поле с URL-адресом в качестве значения атрибута src тега img.

const Person = {

name: 'Gregorio Y. Zara',

theme: {

backgroundColor: 'black',

color: 'pink'

},

url: ‘https://i.imgur.com/7vQD0fPs.jpg’

};

export default function TodoList() {

return (

<div style={person.theme}>

<h1>{person.name}'s Todos</h1>

<img

className="avatar"

src={person.url}

alt="Gregorio Y. Zara"

/>

<ul>

<li>Improve the videophone</li>

<li>Prepare aeronautics lectures</li>

<li>Work on the alcohol-fuelled engine</li>

</ul>

</div>

);

}

***Задание 7.*** Отрегулируйте размер изображения на основе реквизита. В этом примере Avatar получает числовой size реквизит, определяющий <img>ширину и высоту. Измените Avatar компонент, чтобы он запрашивал ближайший размер изображения на основе реквизита size. В частности, если size меньше 90, передайте в функцию getImageUrl аргумент со значением 's'(«маленький»), иначе 'b' («большой»).

import { getImageUrl } from './utils.js';

function Avatar({ person, size }) {

return (

<img

className="avatar"

src={

if (size < 90) {

getImageUrl(person, 's')

} else {

getImageUrl(person, 'b')

}

alt={person.name}

width={size}

height={size}

/>

);

}

export default function Profile() {

return (

<Avatar

size={40}

person={{

name: 'Gregorio Y. Zara',

imageId: '7vQD0fP'

}}

/>

);

}

***Задание 8.*** Показывать значок ❌ для неупакованных элементов (значением isPacked является false), используя условный оператор ? :

function Item({ name, isPacked }) {

return (

<li className="item">

{name} {(isPacked) ? '✔' : '❌'}

</li>

);

}

export default function PackingList() {

return (

<section>

<h1>Sally Ride's Packing List</h1>

<ul>

<Item

isPacked={true}

name="Space suit"

/>

<Item

isPacked={true}

name="Helmet with a golden leaf"

/>

<Item

isPacked={false}

name="Photo of Tam"

/>

</ul>

</section>

);

}

***Задание 9.*** Исправьте обработчик событий. При нажатии на кнопку фон страницы должен переключаться между белым и черным. Однако при нажатии на нее ничего не происходит. Решить проблему. (Не беспокойтесь о внутренней логике handleClick— с этой частью все в порядке.)

export default function LightSwitch() {

function handleClick() {

let bodyStyle = document.body.style;

if (bodyStyle.backgroundColor === 'black') {

bodyStyle.backgroundColor = 'white';

} else {

bodyStyle.backgroundColor = 'black';

}

}

return (

<button onClick={handleClick}> // без скобок

Toggle the lights

</button>

);

}

***Задание 10.*** Добавьте метод, предотвращающий поведение браузера по умолчанию, перезагрузку страницы

export default function Signup() {

return (

<form>

<input />

<button onClick={(e) => {

alert('Submitting!')

e.preventDefault()

}}>

Send

</button>

</form>

);

}

***Задание 11.*** В соответствии с кодом ниже, если вы нажмете на любую из кнопок, сначала сработает ее onClick, а затем onClick родительской <div>. Остановите распространение события.

export default function Toolbar() {

return (

<div className="Toolbar" onClick={() => {

alert('You clicked on the toolbar!');

}}>

<button onClick={(e) => {

alert('Playing!')

e.stopPropagation()

}}>

Play Movie

</button>

<button onClick={(e) => {

alert('Uploading!')

e.stopPropagation()

}}>

Upload Image

</button>

</div>

);

}

***Задание 12.*** Оба input имеют обработчики событий onChange, которые пытаются обновить переменные на основе последнего введенного пользователем значения (e.target.value). Однако переменные, похоже, не "запоминают" свои значения между повторными рендерингами. Исправьте это.

export default function Form() {

const {firstName, setFirstName} = useState(“”);

const {lastName, setLastName} = useState(“”);

function handleFirstNameChange(e) {

setFirstName(e.target.value);

}

function handleLastNameChange(e) {

setLastName(e.target.value);

}

function handleReset() {

setFirstName(“”);

setLastName(“”);

}

return (

<form onSubmit={e => e.preventDefault()}>

<input

placeholder="First name"

value={firstName}

onChange={handleFirstNameChange}

/>

<input

placeholder="Last name"

value={lastName}

onChange={handleLastNameChange}

/>

<h1>Hi, {firstName} {lastName}</h1>

<button onClick={handleReset}>Reset</button>

</form>

);

}

***Задание 13.*** Вот небольшая форма, которая должна позволить пользователю оставить отзыв. Когда отзыв отправлен, предполагается отобразить сообщение с благодарностью. Однако при этом происходит сбой с сообщением об ошибке: "Rendered lesser hooks than expected". Можете ли вы заметить ошибку и исправить ее?

import { useState } from 'react';

export default function FeedbackForm() {

const [isSent, setIsSent] = useState(false);

const [message, setMessage] = useState('');

if (isSent) {

return <h1>Thank you!</h1>;

} else {

return (

<form

onSubmit={(e) => {

e.preventDefault();

alert(`Sending: "${message}"`);

setIsSent(true);

}}

>

<textarea

placeholder="Message"

value={message}

onChange={(e) =>

setMessage(e.target.value)

}

/>

<br />

<button type="submit">Send</button>

</form>

);

}

}