class TemperatureInput extends React.Component {

constructor(props) {

super(props);

this.handleChange = this.handleChange.bind(this);

this.state = {temperature: ''}; }

handleChange(e) {

this.setState({temperature: e.target.value}); }

render() {

const temperature = this.state.temperature; // ...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

render() {

// Ранее было так: const temperature = this.state.temperature;

const temperature = this.props.temperature; // ...

handleChange(e) {

// Ранее было так: this.setState({temperature: e.target.value});

this.props.onTemperatureChange(e.target.value); // ...

class TemperatureInput extends React.Component {

constructor(props) {

super(props);

this.handleChange = this.handleChange.bind(this);

}

handleChange(e) {

***this.props.onTemperatureChange(e.target.value); }***

render() {

***const temperature = this.props.temperature;***

const scale = this.props.scale;

return (

<fieldset>

<legend>Введите градусы по шкале {scaleNames[scale]}:</legend>

<input value={**temperatur**e} onChange={**this.handleChange**} />

</fieldset>

);

}

}

class Calculator extends React.Component {

constructor(props) {

super(props);

this.handleCelsiusChange = this.handleCelsiusChange.bind(this);

this.handleFahrenheitChange = this.handleFahrenheitChange.bind(this);

**this.state = {temperature: '', scale: 'c'}; }**

***handleCelsiusChange(temperature) {***

***this.setState({scale: 'c', temperature}); }***

handleFahrenheitChange(temperature) {

this.setState({scale: 'f', temperature}); }

render() {

const scale = this.state.scale;

**const temperature = this.state.temperature;**

const celsius = scale === 'f' ? tryConvert(temperature, toCelsius) : temperature; const fahrenheit = scale === 'c' ? tryConvert(temperature, toFahrenheit) : temperature;

return (

<div>

<TemperatureInput scale="c"

temperature={celsius} **onTemperatureChange={this.handleCelsiusChange}** />

<TemperatureInput scale="f" temperature={fahrenheit} onTemperatureChange={this.handleFahrenheitChange} />

<BoilingVerdict celsius={parseFloat(celsius)} />

</div>

);

}

}