### **Тест по React (Темы: Components, Props, useState, useEffect)**

**📌 Формат:  
✅ 20 теоретических вопросов (тест + открытые вопросы)  
✅ 5 практических заданий (необходимо предоставить ссылку на GitHub)  
🕒 Время выполнения: 80 минут**

## **📖 Теоретическая часть (20 вопросов)**

### **1. Что такое компонент в React?**

**🔘 a) Функция, которая возвращает JSX  
🔘 b) Файл со стилями  
🔘 c) Любая JavaScript-функция  
🔘 d) Специальный метод render()**

### **2. Как правильно передавать props в компонент?**

**jsx**

**КопироватьРедактировать**

**function Greeting(?) {**

**return <h1>Привет, {name}!</h1>;**

**}**

**🔘 a) {props.name}  
🔘 b) {this.props.name}  
🔘 c) {name}  
🔘 d) {props}**

### **3. Чем useState отличается от обычной переменной?**

**✍ Ответ: Это состояние которое сохраняется даже при рендере страницы**

### **4. Как можно обновить состояние, используя useState?**

**🔘 a) setState(newValue);  
🔘 b) setState(prev => prev + 1);  
🔘 c) state = newValue;  
🔘 d) setState(value => value + 1);**

### **5. Когда вызывается useEffect(() => {...}, [])?**

**🔘 a) При каждом рендере  
🔘 b) Только при монтировании   
🔘 c) Только при размонтировании  
🔘 d) При каждом изменении состояния**

### **6. Как передать props в функциональный компонент?**

**✍ Ответ: {props.name}**

### **7. Что делает return () => {...} внутри useEffect?**

**✍ Ответ: это функция очистки которая срабатывает перед разомантивирования**

### **8. Какие зависимости useEffect нужно передавать, если мы хотим, чтобы он срабатывал только при изменении count?**

**jsx**

**Копировать Редактировать**

**useEffect(() => {...}, [?])**

**✍ Ответ: нужно передать саму переменную зависимости, в нашем случаае [count]**

### **9. Как можно остановить эффект useEffect, работающий с setInterval?**

**✍ Ответ: я+чатжпт: функция очистки ClearInterval()**

### **10. Можно ли обновлять props внутри дочернего компонента? Почему?**

**✍ Ответ: Нет, props переходит сверху вниз. Для изменения props нужно менять родительский класс**

### **11. В чем разница между useState и useRef?**

**✍ Ответ: useState при изменение обновляет компонент то есть рендерит а useRef не нуждается в перересовки**

### **12. В каком порядке выполняются эффекты, если у нас два useEffect в одном компоненте?**

**✍ Ответ: они рендеряться в порядке обьявлении и очереди**

### **13. Какой метод используется для передачи состояния из дочернего компонента в родительский?**

**🔘 a) useState  
🔘 b) callback-функция  
🔘 c) useEffect  
🔘 d) useRef**

### **14. Почему важно указывать зависимости в useEffect?**

**✍ Ответ: это как кодовое слово которое будет помогать контролировать рендер и не давать вечно перезапускаться избегая багов и циклов**

### **15. Можно ли использовать useEffect без зависимостей? Что произойдет?**

**✍ Ответ: вообще можно но не желательно так как эффект будет срабатывать каждый рендер**

### **16. Как можно передать обработчик клика (onClick) в дочерний компонент?**

**✍ Ответ: нужно передавать как свойство пропс из родительского**

### **17. Как правильно передать несколько props в компонент?**

**🔘 a) <Component {title="React"} {subtitle="Hooks"} />  
🔘 b) <Component title="React" subtitle="Hooks" />  
🔘 c) <Component title="React"; subtitle="Hooks"; />  
🔘 d) <Component props={{ title: "React", subtitle: "Hooks" }} />**

### **18. Какой хук React позволяет отслеживать изменения состояния?**

**🔘 a) useEffect  
🔘 b) useState  
🔘 c) useMemo  
🔘 d) useReducer**

### **19. Как правильно использовать useEffect для загрузки данных из API при монтировании компонента?**

**✍ Ответ: можно как исходные данные взять пустой массив внутри useEffect. То есть запрос придет только один раз**

### **20. Какой основной принцип работы с useState при обновлении состояния на основе предыдущего значения?**

**✍ Ответ: обновлять значение основываясь на пред значений, иначе состояние может перемешаться**

## **📝 Практическая часть (5 заданий)**

**📌 Каждое задание нужно выложить в GitHub и оставить ссылку.**

### **1. Создайте компонент Counter, который увеличивает значение при нажатии на кнопку**

**✅ Используйте useState  
✅ Выведите текущее значение**

**Пример работы:**

**css**

**КопироватьРедактировать**

**[+] Кнопка → 1, 2, 3...**

**🔗 Ссылка на GitHub:** **https://github.com/teiimann/ReactStep/tree/main/ProTest/exam**

### **2. Реализуйте компонент UserInfo, который принимает props и отображает имя и возраст**

**✅ Должен получать name и age через props  
✅ Вывести текст вида: "Привет, меня зовут Алекс, мне 25 лет"**

**🔗 Ссылка на GitHub:** **https://github.com/teiimann/ReactStep/tree/main/ProTest/exam**

### **3. Сделайте компонент Timer, который запускает таймер при нажатии на кнопку**

**✅ Используйте useState и useEffect  
✅ Таймер должен останавливаться при размонтировании компонента  
✅ Кнопка "Старт" запускает таймер**

**🔗 Ссылка на GitHub:** **https://github.com/teiimann/ReactStep/tree/main/ProTest/exam**

### **4. Реализуйте компонент DataFetcher, который загружает данные с JSONPlaceholder API**

**✅ Используйте useEffect для запроса  
✅ Выведите загруженные данные**

**Пример API: https://jsonplaceholder.typicode.com/posts/1**

**🔗 Ссылка на GitHub:** **https://github.com/teiimann/ReactStep/tree/main/ProTest/exam**

### **5. Реализуйте компонент Form, который содержит input и кнопку "Отправить"**

**✅ Используйте useState для хранения значения input  
✅ При отправке выводите введенное значение в alert**

**🔗 Ссылка на GitHub:** **https://github.com/teiimann/ReactStep/tree/main/ProTest/exam**