40. Хуки в React

Цель:

Познакомиться со следующими понятиями и возможностями react:

- что такое хуки
- основные хуки react

План занятия:

- Что такое хуки и в чем их особенность
- Разбор основных хуков
    - useState
    - useEffect
    - useCallback
    - useMemo
- Правила хуков

Конспект:

Хуки:

Хуки — нововведение в React 16.8, которое позволяет использовать состояние и другие возможности React без написания классов.

Особенности хуков:

- Хуки позволяют вам повторно использовать логику состояния, не затрагивая дерево компонентов. Благодаря этому, хуки легко использовать в разных компонентах и делиться ими с сообществом
- В классовых сложных компонентах часто можно было встретить смесь не связанной логики. Что приводило к сложной поддержке кода и наличию багов. Чтобы решить эту проблему, хуки позволяют разбить один компонент на маленькие функции по их назначению (например, подписке или загрузке данных), а не на основе методов жизненного цикла.
- Классовые компоненты были одним из барьеров для изучения реакта
- react с использованием хуков еще больше приближает нас к “функциональному” подходу

Хуки — это функции, с помощью которых вы можете «подцепиться» к состоянию и методам жизненного цикла React из функциональных компонентов. Хуки не работают внутри классов — они дают вам возможность использовать React без классов.

Хук состояния. useState

React **useState** Hook позволяет нам отслеживать состояние в функциональном компоненте.

Состояние обычно относится к данным или свойствам, которые необходимо отслеживать в приложении.

React **useState** Hook принимает начальное состояние и возвращает два значения:

- Текущее состояние.
- Функция, обновляющая состояние.

Хук эффекта. useEffect

React **useEffect** Hook позволяет выполнять побочные эффекты в ваших компонентах. Некоторые примеры побочных эффектов: выборка данных, прямое обновление DOM и таймеры.

useEffect принимает два аргумента. Второй аргумент не обязателен.

useEffect(<function>, <dependency>)

<function> - функция, которая будет отрабатывать в зависимости от вида 2 аргумента (массива зависимостей)

<dependency> - массив зависимостей

Работа хука в зависимости от массива dependency:

1. **Зависимости не переданы:** функция будет выполняться при каждом рендере.
2. **Пустой массив:** хук отработает только после первого рендера.
3. **Массив зависимостей:** хук будет отрабатывать
    a. при первом рендере
    b. при изменении значений массива депенденси

Очистка эффекта

Некоторые эффекты требуют очистки, чтобы уменьшить утечку памяти. Таймеры, прослушиватели событий и другие эффекты, которые больше не нужны, должны быть удалены. Мы делаем это, добавляя функцию возврата в конец useEffect.

Хук useCallback

Возвращает мемоизированный колбэк. Передайте встроенный колбэк и массив зависимостей. Хук **useCallback** вернёт мемоизированную версию колбэка, который изменяется только, если изменяются значения одной из зависимостей. Это полезно при передаче колбэков оптимизированным дочерним компонентам, которые полагаются на равенство ссылок для предотвращения ненужных рендеров (например, shouldComponentUpdate).

useCallback(fn, deps) — это эквивалент useMemo(() => fn, deps).

Хук useMemo

Возвращает мемоизированное значение.

Данный хук позволяет улучшать производительность. useMemo работает только тогда , когда один из его депенденси меняет значение.

Правила хуков

- Хуки следует вызывать только на верхнем уровне. Не вызывайте хуки внутри циклов, условий или вложенных функций.
- Хуки следует вызывать только из функциональных компонентов React. Не вызывайте хуки из обычных JavaScript-функций. Есть только одно исключение, откуда можно вызывать хуки — это ваши пользовательские хуки. Мы расскажем о них далее.