

## Сравнение императивного и декларативного подходов к UI

В данной лабораторной работе одно и то же поведение пользовательского интерфейса было реализовано двумя способами: с использованием императивного подхода на jQuery и декларативного подхода на React. Цель работы — понять ключевые различия между этими подходами и причины, по которым декларативная модель React лучше подходит для современных интерфейсов.

В императивной реализации с jQuery разработчик напрямую управляет DOM. Для кнопки вручную добавляется обработчик клика, и при каждом нажатии код явно указывает, какие действия нужно выполнить: добавить или удалить CSS-класс у параграфа с помощью методов вроде `toggleClass`. Такой подход требует пошагового описания всех изменений интерфейса. Логика тесно связана с конкретными DOM-элементами, из-за чего при росте приложения код становится сложнее поддерживать и расширять. Любое усложнение UI приводит к увеличению количества условий и ручных манипуляций с DOM.

Декларативный подход, используемый в React, основан на описании того, *как должен выглядеть интерфейс* при определённом состоянии данных. В компоненте используется состояние (например, `isHighlighted`), которое определяет, должен ли параграф иметь класс подсветки. JSX-разметка описывает UI как функцию от состояния. При нажатии на кнопку изменяется состояние, после чего React автоматически выполняет повторный рендер компонента. Прямого взаимодействия с DOM не происходит.

Как описано в главе 1 книги («Declarative UI structures» и «Data changes over time»), React воспринимает интерфейс как результат текущих данных. Когда данные меняются, React с помощью виртуального DOM сравнивает старое и новое представление интерфейса и применяет только необходимые изменения (diffing и patching).

Декларативный подход лучше масштабируется, так как при усложнении интерфейса разработчику не нужно вручную управлять каждым изменением DOM. Логика становится более предсказуемой, код — чище, а поддержка и развитие приложения — проще.