АО «АЛМАТИНСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»
https://atu.edu.kz/

Факультет «Информационные технологии»
Кафедра «Компьютерная инженерия»

Название курса: «Инструментальные средства разработки программ»



Зиятбекова Гулзат Зиятбеккызы – PhD, ассоциированный профессор
g.ziyatbekova@atu.edu.kz

ЛЕКЦИЯ 14 · ПРОГРАММНЫЕ ИНТЕРФЕЙСЫ

Создание интерфейса программы



От сценария пользователя к состоянию, событию и проверке

01 · ОСНОВА

Интерфейс — это путь пользователя к результату



Пользователь видит

Элементы, состояния и обратную связь.



Программа обрабатывает

Действия и изменения состояния.



Путь становится яснее

Хороший UI снижает неопределённость на каждом шаге.

Интерфейс проектируют как систему сценариев, а не как набор красивых экранов.

02 · ПРИНЦИПЫ

Базовые принципы интерфейса

Понятность

Элемент выглядит так, как работает.

Предсказуемость

Одинаковые действия дают одинаковый результат.

Обратная связь

Система сообщает, что произошло.

Согласованность

Термины, цвета и поведение не конфликтуют.

ДО

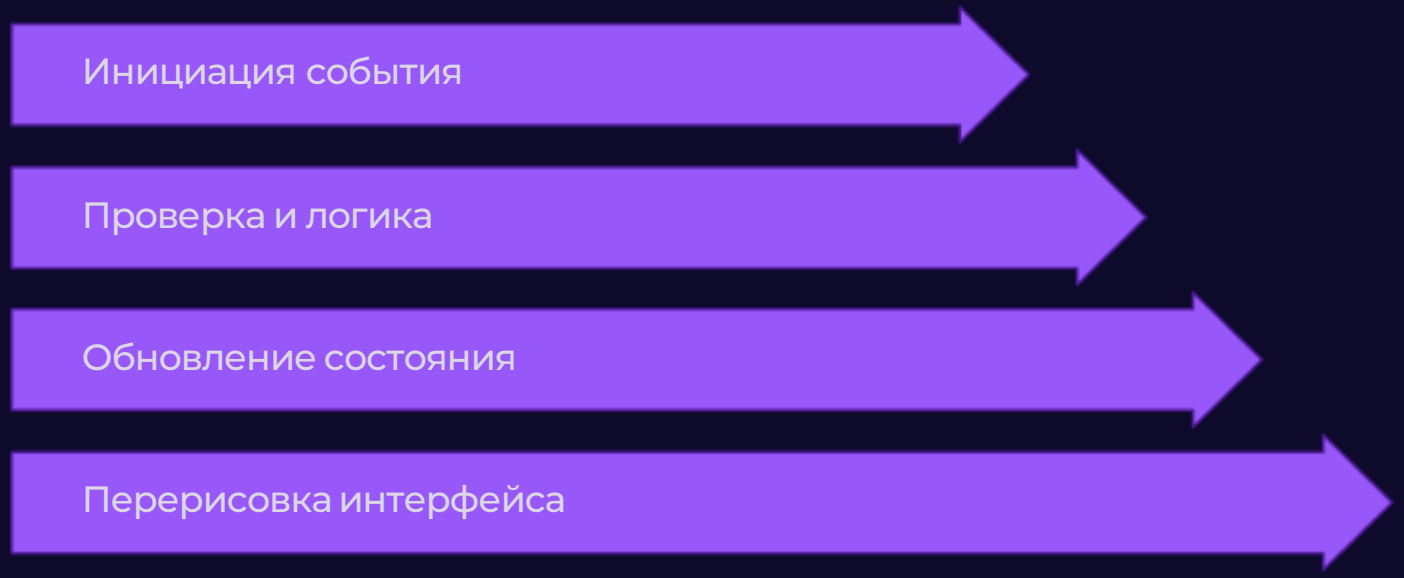
Поле подсвечено, но причина отказа неясна.

ПОСЛЕ

«Введите рабочий адрес электронной почты».

Элемент управления связывает действие и обработчик

Пользователь вызывает click, input, submit или keydown; обработчик проверяет данные и запускает логику, обновляет состояние программы, а интерфейс показывает результат, ошибку или ожидание.



Событие → обработчик → состояние → перерисовка

04 · СОСТОЯНИЕ

Состояние — источник правды интерфейса

Описывайте переходы явно и не дублируйте одно значение в независимых местах.



Отдельно учитывайте данные, загрузку, ошибку, пустые, пограничные и повторные действия.

05 · ВВОД

Обрабатывайте разные способы ввода

- Мышь, клавиатура, сенсорный экран и вспомогательные технологии
- Фокус должен быть видимым и логичным
- Клавиатурное управление нельзя заменять только обработчиком мыши
- Учитывайте разные устройства и размеры экрана

В веб-интерфейсах нужно поддерживать несколько способов ввода, включая мышь, клавиатуру и касание



Мышь

Клик и наведение.



Касание

Жест и размер цели.



Клавиатура

Фокус и Enter.

Технологии доступа

Семантика и озвучивание.

06 · ДОСТУПНОСТЬ

Доступность — часть архитектуры, а не финальная проверка

Семантика

Используйте элементы по их назначению и задавайте доступные имена.

ARIA по необходимости

Роли, состояния и свойства дополняют семантику, а не маскируют её отсутствие.

Фокус и порядок

Навигация с клавиатуры должна идти предсказуемо.

Сообщения

Ошибки и текстовые альтернативы должны быть понятны без цвета и изображения.

W3C связывает доступность с семантикой, клавиатурной поддержкой и доступными именами элементов.

Методы построения интерфейсов

Императивный

Описываем, как менять экран шаг за шагом.

Сценарий: найти элемент и изменить его.

Декларативный

Описываем, каким должен быть UI для состояния.

Сценарий: состояние задаёт отображение.

Компонентный

Собираем интерфейс из переиспользуемых частей.

Сценарий: компонент принимает свойства.

Прототипирование

Проверяем сценарий до дорогой реализации.

Сценарий: ошибка находится раньше кода.

Один интерактивный элемент можно описать разными способами. Выбор подхода меняет не цель, а способ управления UI.

Инструментальные средства разработчика



Figma или аналог

Сценарии, прототипы и визуальная система.



IDE и DevTools

Код, DOM, стили, события и ошибки.



Storybook

Изолированная разработка компонентов.



Автотесты и Lighthouse

Проверка поведения и доступности.

Инструменты должны поддерживать весь цикл: идея → прототип → компонент → тест → наблюдение.

09 · ПРАКТИКА

Рабочий алгоритм создания интерфейса

Двигайтесь от результата пользователя к проверяемому поведению.



Финальная карта: сценарий → состояние → компонент → событие → проверка