

김세아(Seah Kim) ★프리코스 오픈미션

프리코스 과제

2025.10.14 - 2025.10.21

1주차. 문자열 덧셈기 ↗

2025.10.21 - 2025.10.28

2주차. 자동차 경주 ↗

2025.10.28 - 2025.11.04

3주차. 로또 생성기 ↗

2025.11.04- 2025.11.25

오픈 미션 ↗

오픈미션 주제 및 목표

주제

프리코스의 세 가지 과제를 웹 기반 애플리케이션으로 통합 구현

목표

- 핵심 로직(Domain Logic)을 웹 환경에 재사용할 수 있도록 구조화
- 객체 지향 설계(OOP) 원칙을 적용하여 세 개의 독립된 기능을 모듈화하고 통합
- 사용자 친화적인 웹 인터페이스(HTML/CSS) 및 비동기 이벤트 처리를 구현
- 테스트 주도 개발(TDD) 방법론을 준수하여 단위 테스트를 작성

진행 과정

구조 설계

세 도메인이 서로 충돌하지 않고 확장 가능한 형태를 유지할 수 있도록 레이어를 분리한 웹 아키텍처를 선택하였습니다. 객체 지향 설계 원칙을 유지하면서도 모듈별 책임이 명확하도록 구조화하고자 했습니다.

새로운 제약 정의

웹 UI 특성과 실제 서비스 기준을 반영하여 다음과 같은 제약을 명확히 재정의했습니다.

- 로또는 최대 5,000원까지 구매 가능
- 자동차 경주는 최대 10대, 22회 제한

이러한 제약을 도입함으로써 현실적인 유스케이스에 가까운 도메인 설계를 목표로 하였습니다.

개발 환경 구축

Storybook을 활용하여 재사용 가능한 UI 컴포넌트(Header, Footer, LottoBall 등)를 독립적으로 개발 및 문서화할 수 있는 환경을 구성했습니다

도메인 통합

기존 콘솔 환경에서 작성된 Calculator, Lotto, RacingCar 등의 핵심 로직을 features/domain 계층으로 이식하고, Jest 테스트를 웹 환경에 맞춰 수정하며 도메인 무결성을 유지했습니다. TDD 원칙을 따라 테스트를 먼저 작성함으로써 기능 안정성을 확보했습니다.

View 구축

각 미션 페이지(View)는 순수한 UI 렌더링에만 집중하도록 기능을 분리했습니다. Widgets는 HTML 문자열만 반환하며, UI 요소는 Controller에 의존하지 않도록 단일 책임 원칙(SRP)을 엄격하게 적용했습니다.

Controller 개발

CalculatorController, LottoController 등을 구현하여 다음과 같은 책임을 맡겼습니다.

- UI 이벤트 처리
- 입력 유효성 검증 (예: 연속 구분자 방지, 금액 상한 검사)
- 도메인 호출 및 상태 업데이트 관리

컨트롤러 중심 흐름을 통해 도메인과 UI 간의 결합도를 낮추고 유지 보수성을 확보했습니다.

느낀점

1. 스토리북 첫 시도에 대한 느낌

기존에는 전체 애플리케이션을 실행한 상태에서만 UI를 테스트했지만, 스토리북을 사용하니 각 컴포넌트를 완전히 독립된 환경에서 확인할 수 있어서 개발 효율성이 확실히 올라갔습니다. 컴포넌트 별로 다양한 상태를 시각적으로 테스트할 수 있고, args와 argTypes로 문서화를 자연스럽게 함께 진행할 수 있어 재사용성 높은 UI 설계가 가능하다는 점이 인상적이었습니다. 앞으로 자주 사용해보려 합니다!

2. 프레임워크 없이 개발한 느낌

어떤 도전을 해볼까 고민을 하다가, 리액트 같은 프레임워크에 익숙해져 있었기에 바닐라 자바스크립트로 개발하기라는 목표를 세웠습니다. 컨트롤러를 중심으로 DOM 관리, 이벤트 바인딩, UI 업데이트를 모두 수동으로 처리하면서, 프레임워크가 대신 해주던 추상화가 실제로 어떤 역할을 하고 있는지 직접 관찰할 수 있었습니다! 인스턴스 기반 렌더링 구조나 모듈 분리 방식도 직접 고민해야 했기에… 개발 과정에서 자바스크립트 동작 원리와 객체 지향 설계의 기본기를 깊이있게 이해할 수 있었습니다.

UI·도메인·상태·이벤트 구조를 모두 바닐라 JS로 짜야 했기 때문에 어려운 부분도 많았지만, 웹 애플리케이션의 바닥부터 쌓아가는 구조를 실제로 체험할 수 있어서 의미있는 시간이었습니다.