



Chapter 1. OLPP의 기초와 심화

- OLPP의 기본 개념과 역사적 배경
- 객체지향 프로그래밍(OOP)과의 차이점 및 장점
- OLPP의 핵심 원칙: 객체, 링크, 속성, 네트워크
- 복잡한 시스템 설계에서의 OLPP 활용 방법
- 실제 산업에서의 OLPP 적용 사례 연구
- **TOOL: 종이와 연필 - OLPP 구조 직관적 스케치 및 설계**
 - 화이트보드를 활용한 팀 브레인스토밍 기법
 - 마인드맵으로 OLPP 개념 시각화하기
 - 종이 프로토타이핑의 장점과 현대적 활용
- 실습: 간단한 OLPP 시스템 설계 및 다이어그램 작성

Chapter 2. OLPP를 활용한 객체 설계

- 객체 간 관계 모델링의 새로운 접근법
 - 상속 vs 조합 vs 위임의 OLPP적 해석
 - 느슨한 결합과 동적 바인딩의 구현
- 동적 속성 관리와 링크 설정 전략
 - 런타임 속성 변경 및 관리 기법
 - 다중 링크 시스템 설계
- 기존 설계 패턴의 OLPP적 재해석
 - 싱글톤, 팩토리, 옵저버 패턴의 변형
 - 커맨드, 전략, 상태 패턴의 OLPP 구현
- 리팩토링 사례 및 프로세스
- **TOOL: Android Studio - 안드로이드 플랫폼의 설계 패턴 활용**
 - 리팩토링 기능을 활용한 코드 구조 개선
 - 라이브 템플릿을 활용한 OLPP 패턴 적용

- 안드로이드 아키텍처 컴포넌트와 OLPP 통합
- 실습: 기존 애플리케이션의 OLPP 기반 리팩토링

Chapter 3. 자료구조와 OLPP

- 전통적 자료구조의 OLPP적 재정의
 - 링크드 리스트: 객체와 링크의 원형
 - 트리 구조: 계층적 관계의 OLPP 표현
 - 그래프 알고리즘: 네트워크 중심 사고
- 동적 변형 가능한 자료구조 설계
 - 런타임에 구조가 변경되는 시스템 설계
 - 멀티스레드 환경에서의 안전한 자료구조
- 알고리즘 최적화의 OLPP적 접근
 - 링크 기반 캐싱 전략
 - 분산 환경에서의 데이터 접근 최적화
- **TOOL: IntelliJ IDEA - 자료구조 구현 및 테스트**
 - 디버깅 도구를 활용한 자료구조 시각화
 - 성능 프로파일링을 통한 알고리즘 최적화
 - 단위 테스트로 자료구조 신뢰성 검증
- 실습: OLPP 기반 커스텀 자료구조 구현 및 성능 테스트

Chapter 4. 웹 개발에서의 OLPP

- 클라이언트-서버 관계의 OLPP적 모델링
 - 상태 관리와 통신 프로토콜 설계
 - 양방향 데이터 흐름 구현
- RESTful API의 OLPP적 해석과 설계
 - 리소스 간 관계 정의 및 링크 표현
 - HATEOAS 원칙의 실제 적용
- 프론트엔드 컴포넌트의 OLPP적 구조화

- 컴포넌트 간 통신 및 상태 공유
- 마이크로 프론트엔드 아키텍처
- 실시간 웹 애플리케이션 설계
 - WebSocket을 활용한 링크 기반 실시간 통신
 - 이벤트 소싱 및 CQRS 패턴
- **TOOL: WebStorm - 웹 컴포넌트 개발 및 관계 관리**
 - **Swagger/Postman - API 설계 및 테스트**
 - 프론트엔드 프레임워크(React, Vue) 통합
 - TypeScript를 활용한 타입 안전성 확보
- 실습: OLPP 기반 SPA(Single Page Application) 개발

Chapter 5. 데이터베이스와 OLPP

- 관계형 데이터베이스의 OLPP적 모델링
 - 테이블 간 관계와 제약조건 설계
 - ORM과 OLPP의 통합 접근법
- NoSQL 데이터베이스의 문서 및 그래프 구조
 - 문서 지향 DB의 중첩 구조 활용
 - 그래프 데이터베이스와 OLPP의 시너지
- 분산 시스템에서의 데이터 일관성 유지
 - CAP 이론과 OLPP 설계 고려사항
 - 분산 트랜잭션과 데이터 동기화
- 하이브리드 데이터 아키텍처
 - 다중 데이터베이스 환경에서의 OLPP
 - 폴리글랏 퍼시스턴스 전략
- **TOOL: DataGrip - DB 관계 시각화 및 관리**
 - SQL 최적화 및 실행 계획 분석
 - NoSQL 데이터베이스와의 통합(MongoDB, Redis)

- 데이터베이스 버전 관리 및 마이그레이션
- 실습: 복합 데이터 시스템 설계 및 구현

Chapter 6. 임베디드 시스템과 OLPP

- 하드웨어-소프트웨어 통합 설계의 OLPP적 접근
 - 물리적 컴포넌트와 소프트웨어 컴포넌트 간 링크
 - 리소스 제약 환경에서의 최적화
- 실시간 이벤트 및 반응 시스템
 - 이벤트 기반 아키텍처와 OLPP
 - 인터럽트 처리와 우선순위 관리
- IoT 아키텍처의 OLPP적 설계
 - 디바이스-클라우드 통신 모델
 - 센서 네트워크와 메시 토폴로지
- 임베디드 시스템의 안정성과 내결함성
 - 상태 머신과 회복 메커니즘
 - 원격 모니터링 및 관리
- TOOL: CLion - 임베디드 시스템 개발
 - Arduino IDE/Raspberry Pi OS - 하드웨어 프로그래밍
 - 실시간 데이터 모니터링 및 시각화
 - 저수준 디버깅 및 성능 분석
- 실습: OLPP 기반 스마트 홈 컨트롤러 구현

Chapter 7. 클라우드 컴퓨팅과 OLPP

- 클라우드 자원 간 동적 링크 관리
 - 서버리스 아키텍처와 함수 간 관계
 - 리소스 오케스트레이션과 자동 스케일링
- 마이크로서비스 아키텍처의 OLPP 구현
 - 서비스 디스커버리 및 통신 패턴

- API 게이트웨이 및 서비스 메시
- CI/CD 파이프라인의 OLPP적 모델링
 - 빌드-테스트-배포 파이프라인 자동화
 - 인프라스트럭처 as 코드(IaC)
- 멀티 클라우드 및 하이브리드 클라우드 전략
 - 클라우드 간 리소스 연결 및 관리
 - 클라우드 벤더 중립적 설계
- **TOOL: IntelliJ IDEA Ultimate - 클라우드 통합 개발**
 - **AWS/Azure/Google Cloud 콘솔 - 클라우드 자원 관리**
 - **Docker Desktop/Kubernetes Dashboard - 컨테이너 오케스트레이션**
 - 클라우드 리소스 모니터링 및 비용 최적화
- 실습: OLPP 기반 마이크로서비스 시스템 설계 및 배포

Chapter 8. 인공지능과 OLPP

- 신경망 및 머신러닝 파이프라인의 관계 기반 설계
 - 모델 훈련-검증-배포 파이프라인
 - 데이터 전처리 및 특성 엔지니어링 워크플로우
- AI 모델의 투명성과 설명 가능성 분석
 - 블랙박스 모델의 OLPP적 해석
 - 모델 해석 및 의사결정 추적
- 분산 머신러닝 시스템 설계
 - 모델 병렬화 및 분산 훈련
 - 파라미터 서버 아키텍처
- 강화학습 환경의 OLPP적 구현
 - 에이전트-환경 상호작용 모델링
 - 보상 시스템 및 정책 최적화
- **TOOL: PyCharm - 머신러닝 모델 개발 및 시각화**

- **FastAPI - 모델 서빙 및 백엔드 구현**
- **WebStorm/React - 커스텀 모델 모니터링 대시보드 구현**
- **D3.js - 데이터 시각화 컴포넌트 개발**
- 실습: 커스텀 모델 모니터링 시스템 구현

Chapter 9. 프로젝트 관리와 OLPP

- 팀원 및 작업 간 관계의 OLPP적 모델링
 - 역할 기반 책임 및 의존성 관리
 - 크로스 펄서널 팀 구조 설계
- 애자일 방법론의 OLPP적 재해석
 - 스프린트 계획 및 작업 분배 최적화
 - 피드백 루프 및 지속적 개선
- 버전 관리 시스템의 OLPP적 활용
 - 브랜치 전략 및 병합 관리
 - 코드 리뷰 프로세스 최적화
- 지식 관리 및 문서화 전략
 - 기술 문서의 링크 기반 구조화
 - 코드-문서 연계 시스템
- **TOOL: YouTrack - 프로젝트 관리, TeamCity - CI/CD, Space - 협업**
 - **Git/GitHub - 버전 관리 및 협업**
 - **Slack/Microsoft Teams - 팀 커뮤니케이션**
 - 성과 지표 모니터링 및 시각화
- 실습: OLPP 기반 프로젝트 관리 시스템 구축

Chapter 10. OLPP의 미래와 확장

- OLPP의 한계와 도전 과제
 - 대규모 시스템에서의 복잡성 관리
 - 성능 및 확장성 이슈 해결 방안

- 신기술과의 통합 및 시너지
 - 블록체인과 스마트 계약의 OLPP적 모델링
 - 양자 컴퓨팅과 OLPP의 가능성
- 창의적 문제 해결을 위한 OLPP 사고법
 - 시스템 사고와 OLPP의 결합
 - 복잡한 비즈니스 문제의 OLPP적 접근
- 지속 가능한 소프트웨어 개발과 OLPP
 - 코드 유지보수성 및 지속적 리팩토링
 - 레거시 시스템의 점진적 현대화
- **TOOL: 종이와 연필 - 아이디어 시각화 및 직관적 설계**
 - **Figma/Miro - 디지털 협업 및 프로토타이핑**
 - 미래지향적 시스템 설계 워크숍 방법론
- 실습: 미래 기술 트렌드를 적용한 OLPP 시스템 설계

Chapter 11. 안드로이드 개발과 OLPP

- 안드로이드 아키텍처의 OLPP적 해석
 - 컴포넌트 생명주기와 관계 관리
 - MVVM, MVI 아키텍처 패턴
- 액티비티와 프래그먼트 간 관계 모델링
 - 화면 간 네비게이션 및 데이터 전달
 - 백 스택 관리 및 상태 보존
- 안드로이드 Jetpack 라이브러리의 OLPP적 활용
 - LiveData, ViewModel, Room의 관계 기반 사용
 - WorkManager와 백그라운드 작업 관리
- 멀티모듈 안드로이드 앱 설계
 - 기능 모듈화 및 의존성 관리
 - 동적 피쳐 모듈 및 즉시 앱

- 안드로이드 성능 최적화의 OLPP적 접근
 - 메모리 관리 및 누수 방지
 - UI 렌더링 최적화
- **TOOL: Android Studio** - 안드로이드 컴포넌트 설계 및 구현
 - **Firebase Console** - 백엔드 서비스 통합
 - **Layout Inspector** - UI 관계 디버깅
 - **Profiler** - 성능 모니터링 및 최적화
- 실습: OLPP 기반 안드로이드 애플리케이션 개발

부록

- OLPP 디자인 패턴 카탈로그
- OLPP 리팩토링 체크리스트
- OLPP 프로젝트 템플릿 및 스타터 코드
- 주요 툴 설정 및 활용 가이드
- 참고 문헌 및 추가 학습 자료

중급편 주제 : OLPP와 Tool-oriented

고급편 주제 : RealTime과 철학