**Domain-Driven Design**

## Tackling Complexity in the Heart of Software

복잡한 도메인(문제 영역)을 이해하고 모델링하고, 공통 언어로 소통하는데 초점을 맞추는 방법론.

- DDD는 모델을 동작하게 만들어 애플리케이션의 문제를 해결한다.[p.62]
  - 지식 탐구 → Ubiquitous Language 사용 → (단일) 모델 생성 → 모델과 구현을 밀접하게 연관시킴
- DDD의 목표는 **기술보다는 도메인에 대한 모델에 집중해 더 나은 소프트웨어를 만들어내는 것이다.**[p.154]

본문 내용 출처 : Eric Evans. 『도메인 주도 설계』. 이대엽(역). 위키북스, 2011.

Domain-Driven Design, Modeling, Design

#### 서문

서문

#### 1부 동작하는 도메인 모델 만들기

Part I: Putting the Domain Model to Work

1부 동작하는 도메인 모델 만들기

#### 2부 모델 주도 설계의 기본 요소

Part II: The Building Blocks of a Model-Driven Design

2부 모델 주도 설계의 기본 요소

#### 3부 더 심층적인 통찰력을 향한 리팩터링

Part III: Refactoring Toward Deeper Insight

3부 더 심층적인 통찰력을 향한 리팩터링

#### 4부 전략적 설계

Part IV: Strategic Design

**Chapter**


- 14 모델의 무결성 유지
- 15 디스틸레이션
- 16 대규모 구조
- 17 전략의 종합


From:
https://ledyx.xyz/ - **Ledyx WIKI**

Permanent link:
**https://ledyx.xyz/doku.php?id=domain-driven_design&rev=1708185367**

Last update: **2026/10/24 07:03**