

# LDTI (랭코드 개발자 유형 지표)

2020-07-04 v1.0

## LDTI란?

Langcode Developer Type Indicator의 약자로, 성격 유형을 분류하는 MBTI처럼 개발자마다 다른 개발의 방식을 4가지 기준에 따른 16가지 유형으로 분류하는 지표입니다.

개발자 스스로 자신의 유형을 진단하고, 자기계발이나 커리어에 활용하는 것은 물론, 다른 개발자와 협력하거나, 팀장으로서 다른 개발자를 관리하는 등의 협업 과정에서 상호간의 이해와 업무효율을 높일 수 있을 것으로 기대합니다.

또한 비개발자 경영자, 관리자들이 개발자의 독특한 특성과 업무 방식을 이해하지 못해 생기는 문제와 갈등을 완화시키는 가이드가 될 수도 있을 것입니다.

## 분류 기준

개발과 관련된 4가지 측면인 설명, 수행, 설계, 구현에 있어 선호하는 방식을 기준으로 사용

**설명 방식:** 개발 과정과 결과물에 대한 설명을 할 때 선호하는 방식

**D – Document:** 기획서, 설계서, 매뉴얼, 상세한 주석 등의 문서를 작성하고 활용

**C – Code:** 코드 그 자체의 내용과 프로그램, 서비스의 모습과 작동과정을 보여줌

**수행 방식:** 실제 개발을 위한 코드 작성을 수행할 때 선호하는 방식

**F – Follow:** 요구사항, 모범사례, 정석, 표준 등의 기준과 규칙을 따르는 것을 우선순위에 높음

**N – Negotiate:** 편의성, 생산성 등을 위해서라면 기준과 규칙은 타협의 대상이 될 수 있다고 봄

**설계 방식:** 개발을 위한 구조, 디자인, 패턴 등의 설계를 수립할 때 선호하는 방식

**S – Sequential:** 설계를 마치고 코드를 작성하며, 변경사항을 수합해 재설계 단계를 거쳐 수정

**P – Parallel:** 변경사항을 그때그때 설계에 반영하며, 설계와 코딩의 단계를 구분하기보단 병행

**구현 방식:** 여러 문제에 대한 솔루션을 구현할 때 선호하는 방식

**O – One big solution:** 여러 문제에 유연하게 대응할 수 있는 큰 솔루션 하나를 구현

**M – Many small solutions:** 개별 문제에 대응하는 작은 솔루션 여럿을 구현

## 유형

16개 유형을 수행, 설계 2개 기준의 조합에 따라 4가지 그룹으로 구분하는 것이 가능합니다.

**FS (모범생):** 정해진 절차와 규칙에 따라 체계적인 방식으로 개발하는 것을 중시

**FP (지휘자):** 정석과 기준에 따른 코드 조각을 필요에 따라 배치하고 조합하여 개발

**NS (요리사):** 구체적이고 명확한 설계와 순서 속에서 자신만의 테크닉을 발휘하여 개발

**NP (탐험가):** 더 높은 편의성, 생산성을 위해 새로운 방식을 찾고 시도하는 것을 중시

16개 유형마다 해당 유형의 개발 과정, 결과물 등을 건축물에 비유하여 명명했습니다.

<b>DFSO</b> 피라미드	<b>DFPO</b> 마천루	<b>DNSO</b> 주상복합	<b>DNPO</b> 복합쇼핑몰
<b>DFSM</b> 전원주택단지	<b>DFPM</b> 타운하우스	<b>DNSM</b> 한옥마을	<b>DNPM</b> 카페거리
<b>CFSO</b> 콜로세움	<b>CFPO</b> 국제공항	<b>CNSO</b> 테마파크	<b>CNPO</b> 카지노
<b>CFSM</b> 대학교	<b>CFPM</b> 산업단지	<b>CNSM</b> 풀빌라	<b>CNPM</b> 먹자골목

개별 유형별 상세 설명은 다음과 같습니다.

### DFSO 피라미드

거대한 피라미드를 짓기 위해 튼튼하고 검증된 석재를 블록으로 삼아 명확한 절차와 규칙, 문서에 따라 쌓아올린 것처럼, 이들의 개발은 놀라울 정도로 체계적이고 안정적입니다.

### DFPO 마천루

튼튼한 기반 위에 서서 하늘을 향해 높이 솟아오른 마천루처럼, 이들의 개발은 안정적인 설계와 까다로운 기준을 맞추면서, 수많은 층마다 다른 용도에 대응하는 수정까지 빠지지 않습니다.

### DNSO 주상복합

주거지와 상가를 합친 주상복합은 까다로운 규제와 변수를 고려해야 하지만, 이들의 개발은 복잡한 기준과 규칙을 충족시키면서도 필요한 모든 기능을 최소의 자원으로 구현합니다.

### DNPO 복합쇼핑몰

수많은 쇼핑객들을 위한 다양한 상가와 편의시설까지 없는 게 없는 복합쇼핑몰에 규칙은 없지만 고객들을 위한 안내는 언제나 충실하듯, 이들의 개발은 다채롭지만 혼란스럽지 않습니다.

## **DFSM** 전원주택단지

1층짜리 작은 집이지만, 어엿한 마당까지 가진 전원주택이 반듯하게 뻗은 길을 따라 가득한 풍경처럼, 이들의 개발은 작지만 신뢰할 수 있는 요소들이 모여 우아한 구조를 만들어냅니다.

## **DFPM** 타운하우스

단독주택 같으면서 아파트의 장점까지 가진 타운하우스처럼, 이들의 개발은 기능마다 최적화된 요소들이 일관적인 기준을 따라 조화롭게 붙어있습니다.

## **DNSM** 한옥마을

역사와 전통이 있으면서도 집집마다 다른 매력이 있는 한옥마을처럼, 이들의 개발은 체계와 규칙을 따르면서도 실용성도 무시하지 않은 요소들이 잘 어우러져 있습니다.

## **DNPM** 카페거리

제각각 다른 매력과 맛을 자랑하는 카페들이 늘어선 거리처럼, 이들의 개발은 각기 다른 요소들이 꼭 늘어섰지만 각 요소들의 특징은 명확하고 뚜렷하게 제시되어 있습니다.

## **CFSO** 콜로세움

로마제국의 질서와 체제를 상징하는 웅장한 콜로세움처럼, 이들의 개발은 큰 규모에도 불구하고 명확한 절차와 기준을 자랑하면서 누구나 그 역할을 알 수 있는 직관성까지 갖추고 있습니다.

## **CFPO** 국제공항

넓은 활주로에, 수많은 사람들을 위한 터미널과 커다란 격납고를 갖춘 국제공항처럼, 이들의 개발은 수많은 요구사항을 훌륭하게 충족시키면서, 그 구조를 한 눈에 알아볼 수 있습니다.

## **CNSO** 테마파크

방대한 부지에 화려한 놀이기구와 편의시설이 꽉 찬 테마파크처럼, 이들의 개발은 다양한 방법으로 구현된 매력적인 기능들을 뽐내지만 그 기저에는 세세하고 정교한 설계가 존재합니다.

## **CNPO** 카지노

화려한 야경, 우아한 호텔, 짜릿한 승부까지 모든 것이 있는 카지노처럼, 이들의 개발은 그 거대한 구석구석마다 비상한 스킬과 신선한 아이디어로 가득해 필요한 모든 것을 빠르게 제공합니다.

## **CFSM** 대학교

캠퍼스에 늘어선 강의실과 건물들 사이로 학문이 숨쉬는 대학교처럼, 이들의 개발은 검증된 설계와 구조에 따른 요소들이 모여 상황마다 다른 요구에 체계적인 해결책을 제공합니다.

## **CFPM** 산업단지

수많은 공장들이 멈추지 않고 각각 다종다양한 제품을 만들어내는 산업단지처럼, 이들의 개발은 역할에 충실한 요소들이 모여 요구사항에 맞는 동작을 일관적으로 수행합니다.

### CNSM 풀빌라

수영장이 딸린 숙소라는 매력적인 풀빌라처럼, 이들의 개발은 과감한 발상과 용의주도한 설계를 따라 요소들을 조합하여 다양한 기능과 함께 그 이상의 시너지를 창출합니다.

### CNPM 먹자골목

확자지결한 거리를 따라 당신이 생각하는 모든 맛집들이 늘어선 먹자골목처럼, 이들의 개발은 당신이 생각한 모든 것에 대응하는 다양한 요소들이 한가득 모여 선택을 기다리고 있습니다.

## 유의사항

같은 개발자라 할지라도 프로젝트의 특성, 기간, 규모나 사용하는 언어, 기술 등에 따라 개발의 방식은 달라지게 됩니다. 작성자 본인은 CNPO(카지노) 유형이지만, 주로 사용하는 C#이 아닌 다른 언어, 특히 python이나 javascript 등을 사용할 때는 CNPM(먹자골목) 유형에 가깝습니다.

따라서 유형을 분류할 때는, 현재 사용하는 언어나 진행 중인 특정한 프로젝트를 개발하는 방식에 근거하는 대신, 언어, 기술, 프로젝트의 상세사항 등에서 개발자인 자신이 100% 선택권을 가지고 있다고 가정하고 판단을 진행할 필요가 있습니다.

각 기준마다 판단이 애매한 경우가 있을 수 있습니다. 이런 경우 D, F, S, O가 좀 더 전통적이고, 교과서나 큰 조직 등에서 볼 법한 느낌이라는 점을 고려하시면 도움이 될 수 있습니다. 구로디지털단지 소재 대형 SI업체의 13년 경력 JAVA 개발자가 부장으로 있는 팀에서 주로 보이는 특징을 생각하시면 어떨까요? 가령 문서를 쓰긴 쓰는데 꼼꼼하게는 안 하고 코드 작성에 더 집중하는 경우, D와 C가 헛갈릴 수 있습니다. "문서는 써야한다!"고 생각하기 때문에 대충이라도 문서를 쓴다면 D로, "쓰면 좋긴 하니" 문서를 작성한다고 하면 C에 해당한다고 판단할 수 있겠습니다.

설명, 수행, 설계, 구현의 4개 기준이든, FS, FP, NS, NP의 4개 그룹이든, 16개 개별 유형이든 다소 불분명하거나 납득하기 힘든 내용이 있을 수 있습니다. 어떤 부분이든 수정할 부분이나 미흡한 점이 있다면 가감없이 피드백 주시면 많은 도움이 됩니다.

## 피드백 및 문의

[admin@langcode.io](mailto:admin@langcode.io)