

性能・安全性・開発体験のバランスを追求した理想的なプログラミング言語の作成

背景と目的

既存言語の課題解決

- C/C++: 脆弱性・管理負担
- Rust: 高学習コスト・長コンパイル
- Python: 速度・型安全性の欠如
- Java: 冗長な記述

プロジェクトの狙い

既存言語の長所を融合した「丁度いい言語」の創出。低レイヤー知識と言語理論の習得を目指す。

言語の特徴

ネイティブ生成

マルチパラダイム

静的型付け

クロスコンパイル

自動メモリ管理 (GC)

GCにより安全性を確保し、ユーザーのメモリ管理の負担を軽減する。

使用技術

Lang: C++ std=20 (Clang)

Backend: LLVM 18

開発目標

プロジェクトリサーチでの目標

- コンパイラの作成
- (標準ライブラリの作成)

最終目標

- インタプリタの開発
- 充実したドキュメント整備
- わかりやすいエラーメッセージ
- パッケージマネージャ・ツールチェーン
- フォーマッタ等周辺ツールの開発