



言語処理系を素早く作るプラットフォーム

あなた言語の作り方

Kray-G, Mr.Diamond Global Blue Publisher
December 3, 2020

目次

第 1 章	はじめに	1
1.1	本書の目的	1
1.1.1	職人気質	1
1.1.2	プログラマーという人種	1
1.2	軽量 JIT コンパイラ MIR	1
第 2 章	設計編	3
2.1	プログラミング言語 KiSS	3
2.2	ミニマム仕様	3
2.3	拡張の方針	3
2.4	言語処理系の動作概要	3
第 3 章	実装編	5
3.1	Parser を作る	5
3.2	Lexer を作る	5
3.3	AST を作る	5
3.4	意味解析を行う	5
3.5	コード出力	5
3.6	サンプルプログラムの実行	5
第 4 章	拡張編	7
4.1	型の追加	7
4.1.1	文字列のサポート	7
4.1.2	多倍長整数のサポート	7
4.1.3	バイナリ配列のサポート	7
4.1.4	配列のサポート	7
4.1.5	連想配列のサポート	7
4.1.6	範囲オブジェクトのサポート	7
4.2	高度な機能の追加	7
4.2.1	クロージャのサポート	7
4.2.2	クラスのサポート	7
4.2.3	モジュールのサポート	7
4.2.4	ファイバーのサポート	7
4.3	制御構造の追加	7
4.3.1	for-in	7
4.3.2	switch-case	7

第 5 章 「あなた言語」を作ろう	9
5.1 ボイラープレートの作成	9
5.2 あなたが作成する部分	9
5.2.1 フロントエンド	9
5.2.2 AST 拡張	9
5.2.3 ライブラリ作成	9
付録 A ソースコード	11
付録 B 謝辞	13

図目次

表目次

1.1 本書の目的

1.1.1 職人気質

さて、プログラミング言語とは全く関係ない話から始まりますがご容赦ください。

ブライアン・メイというギタリストがいます。有名過ぎて説明不要ですが、イギリスのロックバンド Queen のギタリストです。彼は自宅や知人の家の廃材などを使って自分オリジナルのギター「レッド・スペシャル」を作成したことで有名です。このギターは 7 色の音を出せる唯一無二のギターとして知られています。また、エドワード・ヴァン・ヘイレンというギタリストがいます¹。この方も有名過ぎて説明は不要ですが、自らカスタムした「フランケンシュタイン」というギター使っていました。その道のプロは、自らの道具は自分の手に合うように自分専用に作ってしまうものなのです。

1.1.2 プログラマーという人種

現在、プログラマーは何かしらのプログラミング言語を使って仕事をしています。そのプログラミング言語は「誰か」が「何かの問題を解決するために」作ったものです²。プログラミング言語の作り方を解説した本は数多くありますが、サンプル言語の作成を通して「作り方を学ぶ」「コンパイラの仕組みや実装手法を学ぶ」ということが主眼に置かれています。したがって、実用的な自分のためのプログラミング言語を作るのはまた別の話であり、学んだことをベースに自分で 1 から作らなければならないケースが多くありました。

本書は最終的に「あなたが『あなた言語』を作るためのプラットフォームを作ること」を目的に書きました。もちろん、サンプル言語作成を通してプログラミング言語の作り方を学べるようになっていますが、最終的にはそれは実用的なプラットフォームとして機能し、そのまま簡単に「あなた言語」を作れるようにする、ということが最大の目的となります。

もちろん、何を抽象化するかによって基盤として成り立つかが分かれていますが、概ね一般的なプログラミング言語を作成することは可能とする予定です。

1.2 軽量 JIT コンパイラ MIR

本書を書こうとしたもう一つの理由が、この MIR³ の存在です。

MIR は、Ruby の MJIT を改善することが最初の目的として示されていますが、MIR 自体は Ruby には依存しない JIT 環境として設計・実装されています。つまり、MIR をバックエンドに使用することによって、MIR がサポートする様々なプラットフォームで実行可能とすることができます。本書ではこの MIR をバックエンドに使用する予定です。

¹ まことに残念ではありますが 2020 年 10 月 6 日、65 歳の若さで亡くなりました。天才がまた一人...

² 実は単に作りたかった、というケースも多いのですが。

³ <https://github.com/vnmakarov/mir>

これにより、MIR への入力を作り出すフロントエンドを作成するだけで実行可能なプログラミング言語が作成できます。また、フロントエンドの多くの部分をボイラープレート化して用意することで、言語処理系を実装するための様々な苦労を低減できるのではと考えています。

2.1 プログラミング言語 KiSS

本書のゴールは「言語処理系を簡単に作ることができる基盤を構築する」です。そのために以下のサンプル言語作成を通して実際に動作する基盤自体を構築していきます。

- サンプル言語 KiSS
 - KiSS - **K**inx **S**ubset with **S**tatically-typed の略です。

ここで、Kinx という名前が出てきましたが、これは私の作成した動的型付けタイプのプログラミング言語¹です。これも私自身が使いたい言語、というものを想像したとき、世の中には無く自分の手にフィットするようなプログラミング言語が欲しいという動機から始まりました。今回実装するプログラミング言語 KiSS は、この Kinx のサブセットと表現しているものの、文法はある程度近づけますが、静的型付けの言語として新規にデザインする予定です。

2.2 ミニмум仕様

2.3 拡張の方針

2.4 言語処理系の動作概要

¹ <https://github.com/Kray-G/kinx>

- 3.1 **Parser** を作る
- 3.2 **Lexer** を作る
- 3.3 **AST** を作る
- 3.4 意味解析を行う
- 3.5 コード出力
- 3.6 サンプルプログラムの実行

4.1 型の追加

4.1.1 文字列のサポート

4.1.2 多倍長整数のサポート

4.1.3 バイナリ配列のサポート

4.1.4 配列のサポート

4.1.5 連想配列のサポート

4.1.6 範囲オブジェクトのサポート

4.2 高度な機能の追加

4.2.1 クロージャのサポート

4.2.2 クラスのサポート

4.2.3 モジュールのサポート

4.2.4 ファイバーのサポート

4.3 制御構造の追加

4.3.1 for-in

4.3.2 switch-case

第 5 章

「あなた言語」を作ろう

5.1 ボイラープレートの作成

5.2 あなたが作成する部分

5.2.1 フロントエンド

5.2.2 AST 拡張

5.2.3 ライブラリ作成

