

23-Interpreter

Seeeeeee:D 夏休み勉強会

デザインパターン

Katsuya Suzuki, Mingzhe Yang, Nanaka Ishijima

Interpreterパターンとは

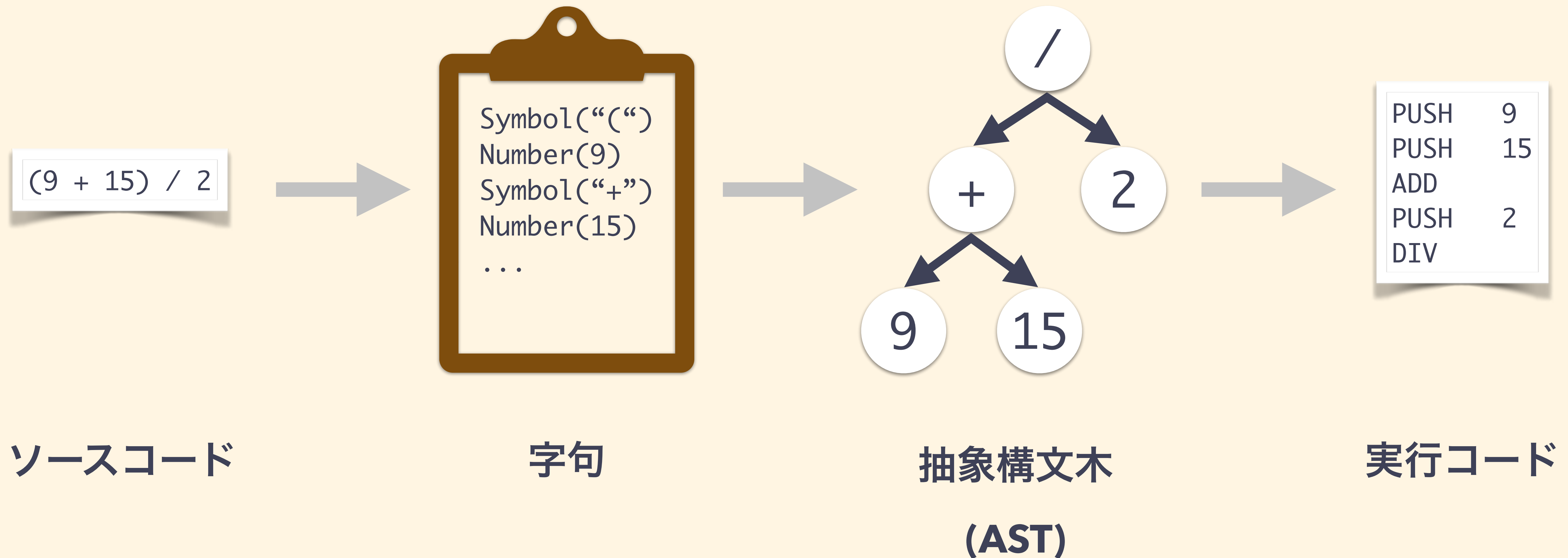
- 特定の文法表現に従って文字列を解釈した結果に対して、値を評価するためのデザインパターン
- 俗に言うインタプリタ(\leftrightarrow コンパイラ)の一部を実装するためのデザインパターン

インタプリタとコンパイラの違い

- コンパイラ: ある言語から別の言語へ変換する
 - C → 機械語 にコンパイル
 - Ruby → C (の内部表現) にコンパイル
- インタプリタ: ある言語を別の表現へ変換し(最終的に)実行する

参考: <https://www.ruby.or.jp/ja/tech/install/ruby/implementations.html>

インタプリタが結果を返すまでの流れ



Compositeパターンと似ている

- 木構造を取るという点で同じ
- 葉と節を区別するかしないかが異なる
 - Composite → ファイル(=葉)とフォルダ(=節)
を区別しない
 - Interpreter → オペランド(=葉)と演算子(=節)
を区別する

動作する様子を確認しよう！

VSCodeへ

まとめ

- インタプリタを実際に実装してみると楽しい&勉強になる
- オレオレ言語を実装できるようになった！
- 用途が限定的すぎる印象がある
- 実装過程で、クラスの責務の分け方は参考になった