PHP で PHP を作る (縮小版)

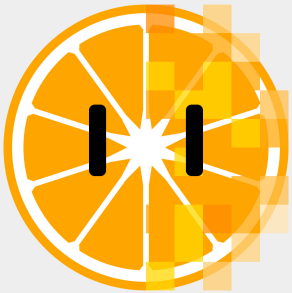
nsfisis (いまむら)

第 169 回 PHP 勉強会@東京

自己紹介

いまむら

nsfisis



@ デジタルサーカス株式会社

PHP で PHP を作る

- 簡単な言語処理系は簡単に作れる
- PHP でも言語処理系を作れる

PHP で PHP を作る

- 簡単な言語処理系は簡単に作れる
- PHP でも言語処理系を作れる
- PHP で、FizzBuzz が動くだけの PHP 処理系を実装してみよう

今回の制約

- なるべく言語処理系特有の知識・専門用語を使わずに実装・説明してみる
- 今回のプログラムがギリギリ動かせるくらいのミニマムで愚直な実装を目指す
- 今回出てこない用語 : 文法クラス、構文解析、AST、VM、バイトコード等

動かすプログラム (FizzBuzz)

```php
<?php
for ($i = 1; $i <= 100; $i++) {
  if ($i % 15 === 0) {
    echo "FizzBuzz";
  } elseif ($i % 3 === 0) {
    echo "Fizz";
  } elseif ($i % 5 === 0) {
    echo "Buzz";
  } else {
    echo $i;
  }
  echo "\n";
}
```

全体の流れ

1. ソースコードという一かたまりの文字列を意味のある最小単位 (単語) に分割する
2. 前から順番に単語を見ていき、それに対応した処理をおこなう

単語への分割

```php
<?php
for ($i = 1; $i <= 100; $i++) {
  if ($i % 15 === 0) {
    echo "FizzBuzz";
  } elseif ($i % 3 === 0) {
    echo "Fizz";
  } elseif ($i % 5 === 0) {
    echo "Buzz";
  } else {
    echo $i;
  }
  echo "\n";
}
```

単語への分割

```php
function split_into_words(string $input): array {
  $i = 0;
  $result = [];
  while ($i < strlen($input)) {
    $first = $input[$i];
    if ($first === '<') {
      // ...
    }
  }
  return $result;
}
```

単語への分割

```
while ($i < strlen($input)) {
  $first = $input[$i];
  if ($first === '<') {

    ...
  } else if ($first === '(') {
    $result[] = Word::LeftParen;
    $i += 1;
  } else if ($first === ')') {
    $result[] = Word::RightParen;
    $i += 1;
  } else if (...) {

    ...
  }
}
```

単語への分割

```
  ...
} else if (ctype_space($first)) {
  $i += 1;
} else if (ctype_digit($first)) {
  $j = $i;
  while (ctype_digit($input[$j])) {
    $j += 1;
  }
  $result[] = (int) substr($input, $i, $j - $i);
  $i = $j;
} else if (...) {

  ...
```

単語への分割

```
  ...
} else if (ctype_alpha($first)) {
  $j = $i;
  while (ctype_alpha($input[$j])) {
    $j += 1;
  }
  $result[] = match (substr($input, $i, $j - $i)) {
    'echo' => Word::Echo_,
    'for' => Word::For_,
    'if' => Word::If_,
    'elseif' => Word::ElseIf_,
    'else' => Word::Else_,
  };
  $i = $j;
} else if (...) {
```

単語への分割

```php
<?php
for ($i = 1; $i <= 100; $i++) {
  if ($i % 15 === 0) {
    echo "FizzBuzz";
  } elseif ($i % 3 === 0) {
    echo "Fizz";
  } elseif ($i % 5 === 0) {
    echo "Buzz";
  } else {
    echo $i;
  }
  echo "\n";
}
```

単語への分割

```
[
  Word::PhpTag,
  Word::For_,
  Word::LeftParen,
  new Variable("i"),
  Word::Assign,
  1,
  Word::Semicolon,

  ...,
]
```

次のステップ

単語の配列を前から順番に見ていき、対応する処理をおこなう

実行する

```
class Php
  private array $words;
  private int $position;
  private array $variables;


  public function __construct(array $words) {
    $this->words = $words;
    $this->position = 0;
    $this->variables = [];
  }
}
```

実行する

```
class Php {

  ...
  public function runPhp() {
    $this->expectWord(Word::PhpTag);
    $this->runStatements();
  }
  private function expectWord(Word $expected_word) {
    if ($this->words[$this->position] !== $expected_word) {
      throw new RuntimeException(...);
    }
    $this->position += 1;
  }
}
```

実行する

```
private function runStatements() {
  while (true) {
    $first = $this->words[$this->position] ?? null;
    if ($first === Word::For_) {
      $this->runForStatement();
    } else if ($first === Word::If_) {
      $this->runIfStatement();
    } else if ($first === Word::Echo_) {
      $this->runEchoStatement();
    } else {
      break;
    }
  }
}
```

実行する

```
private function runEchoStatement() {
  $this->position += 1; // skip 'echo'
  $value = $this->calculateExpression();
  echo $value;
  $this->expectWord(Word::Semicolon);
}
```

実行する

```
private function calculateExpression() {
  $left_hand_side = $this->getNextWord();
  while (true) {
    $next_word = $this->words[$this->position];
    if ($next_word === Word::StrictlyEqual) {
      $this->position += 1; // skip '==='
      $right_hand_side = $this->getNextWord();
      $left_hand_side = $left_hand_side === $right_hand_side;
    } else if (...) {

      ...
    } else {
      return $left_hand_side;
    }
  }
}
```

実行する

```
private function runIfStatement() {
  $this->position += 1; // skip 'if'
  $this->expectWord(Word::LeftParen);
  $condition = $this->calculateExpression();
  $this->expectWord(Word::RightParen);
  $this->expectWord(Word::LeftBrace);
  $this->runStatements();
  $this->expectWord(Word::RightBrace);
}
```

実行する

```
private function runIfStatement(bool $doRun = true) {
  $this->position += 1; // skip 'if'
  $this->expectWord(Word::LeftParen);
  $condition = $this->calculateExpression($doRun);
  $this->expectWord(Word::RightParen);
  $this->expectWord(Word::LeftBrace);
  $this->runStatements($doRun && $condition);
  $this->expectWord(Word::RightBrace);
}
```

実行する

```
private function runEchoStatement(bool $doRun = true) {
  $this->position += 1; // skip 'echo'
  $value = $this->calculateExpression($doRun);
  if ($doRun) {
    echo $value;
  }
  $this->expectWord(Word::Semicolon);
}
```

実行する

```
for ($i = 1; $i <= 100; $i++) {

  ...
}
```

実行する

```
$this->calculateExpression($doRun);
$condition_position = $this->position;
while (true) {
  $condition_result = $this->calculateExpression($doRun);
  $update_position = $this->position;
  if (!$condition_result) {
    $this->calculateExpression(doRun: false);
    $this->runStatements(doRun: false);
    break;
  }
  $this->calculateExpression(doRun: false);
  $this->runStatements($doRun);
  $this->position = $update_position;
  $this->calculateExpression($doRun);
  $this->position = $condition_position;
}
```

まとめ

- 簡単な言語処理系は簡単に作れる
- 今回の実装は 286 行
- 昔の PHP は、これと大して変わらないくらいのアーキテクチャだった

宣伝

PHP カンファレンス小田原 2025 (4/12)

11/4 プロポーザル募集開始！

「匿名プロポーザル」を実施します (詳細は note 記事へ)