

Interpreterとは

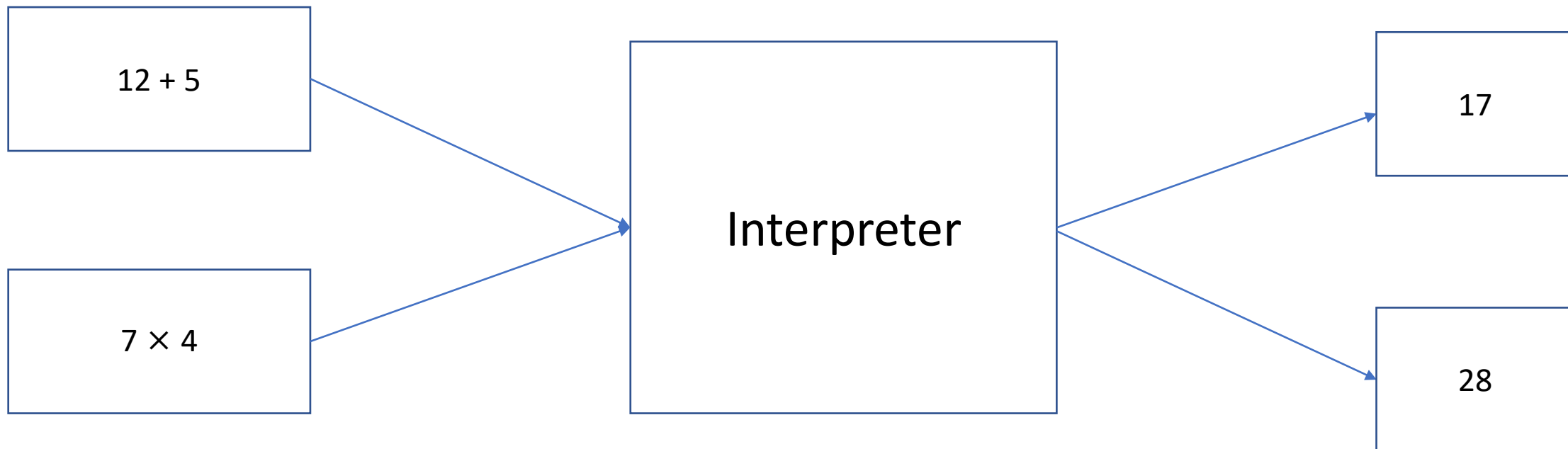
# Interpreterパターン

Interpreterとは翻訳者のこと

SQL、HTML、プログラムなどの一定の規則性を持った文の構文を解析するためのパターン。

文字列から、オブジェクトの型に変換する。

プログラムなどの構文として正しいかチェックする場合などに用いられる。



# Interpreterパターン

## 目的

Compositeパターンの階層構造を用いて、インタプリタを実装する。インタプリタとは、プログラム、SQLなどの構文を解釈するもののこと

## 仕組み

構文解析のための処理を記述したクラスを作成する。

構文解析される情報を持ったクラスを作成して、インスタンス化したのちに構文解析させる

## 構成要素

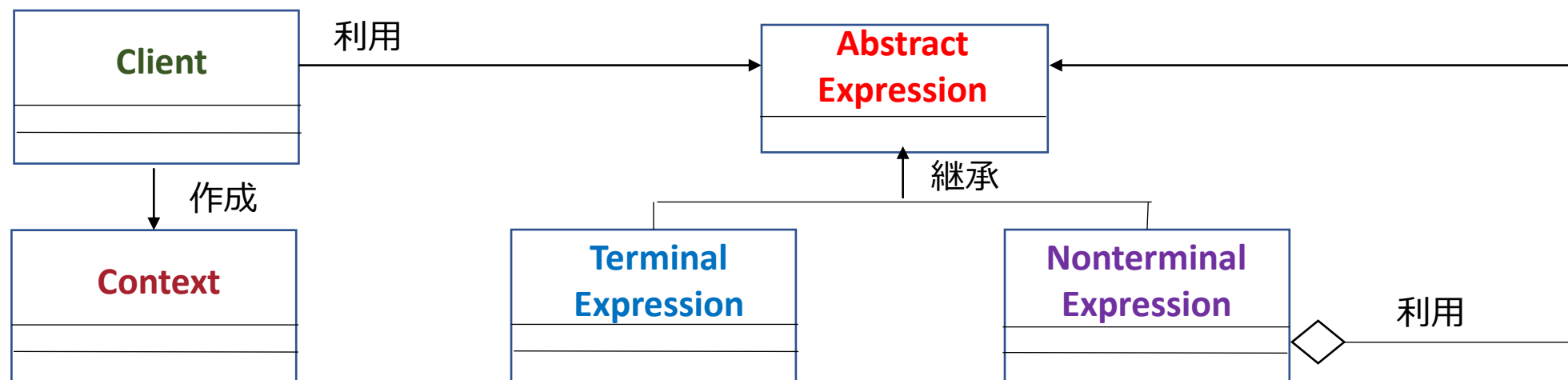
**AbstractExpression**: 構文木の各ノードに共通の処理を記載したインタフェース

**TerminalExpression**: **AbstractExpression**を具体化したクラス。木構造の葉を表す

**NonterminalExpression**: **AbstractExpression**を具体化したクラス。木構造の枝を表す

**Client**: 命令を実行するクラス

**Context**: **AbstractExpression**によって構文解析されるクラス

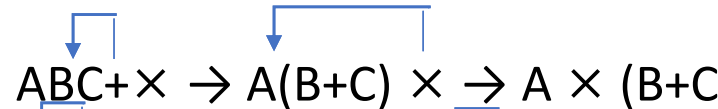
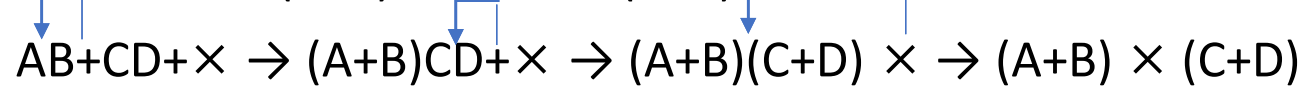


# Interpreterパターン

**逆ポーランド表記法:** 演算子を項の後に書く記法

$$A \times (B + C) \Rightarrow ABC+ \times$$

$$(A + B) \times (C + D) \Rightarrow AB+CD+ \times$$


$$ABC+ \times \rightarrow A(B+C) \times \rightarrow A \times (B+C)$$

$$AB+CD+ \times \rightarrow (A+B)CD+ \times \rightarrow (A+B)(C+D) \times \rightarrow (A+B) \times (C+D)$$

$$1\ 2\ + \Rightarrow 1 + 2 \Rightarrow 3$$

$$1\ 5\ * \ 1\ 2\ + \ + \Rightarrow (1 * 5) + (1 + 2) = 8$$

$$1\ 1\ + \ 3\ 4\ * \ + \ 2\ /\ 3\ 3\ * \ *$$

$$\Rightarrow ((1 + 1) + (3 * 4)) / 2 * 3 * 3 \Rightarrow 63$$