MainClass

+ MainClass()
+ Main(args: String[]) : Void
+ TestCases() : Void

Function

+ Call(function: String, argumentStack: Stack<Token>) : Double
- GetArg(argumentStack: Stack<Token>) : Double
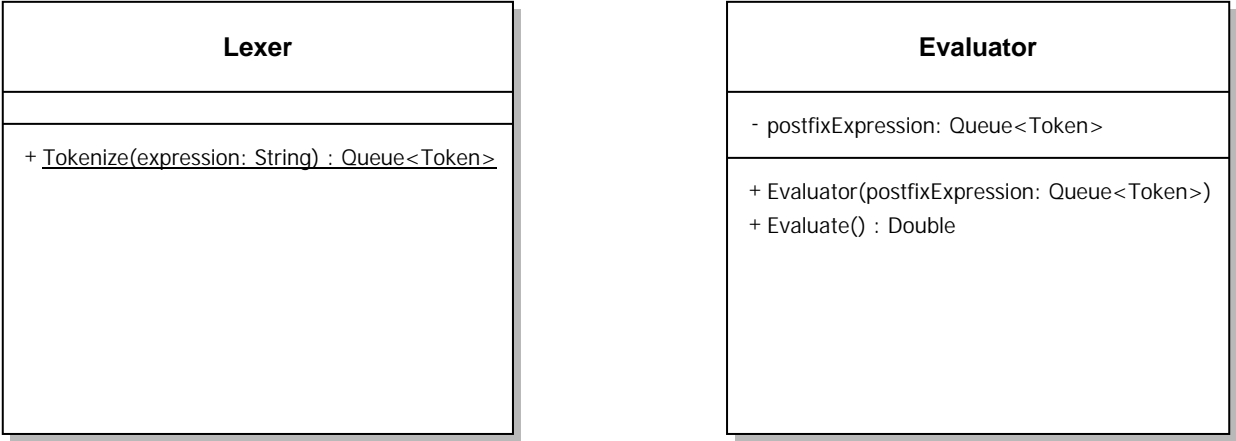
ShuntingYard

- infixTokens: Queue<Token>

+ PostfixTokens { get; } : Queue<Token>
+ ShuntingYard(infixTokens: Queue<Token>)
- CloneQueue(queue: Queue<Token>) : Queue<Token>
- InfixToPostfix() : Queue<Token>

Lexer

+ Tokenize(expression: String) : Queue<Token>

Evaluator

- postfixExpression: Queue<Token>

+ Evaluator(postfixExpression: Queue<Token>)
+ Evaluate() : Double

«enumeration»
TokenType

None = 0
Number = 1
Function = 2
FunctionArgSeparator = 3
Operator = 4
LeftParen = 5
RightParen = 6

Token

- type: TokenType
- value: String
- validTokens: Dictionary<String, TokenType>

+ Type { get; } : TokenType
+ Value { get; } : String
+ Token(value: String)
- Token()
- GetTokenType() : TokenType

1   type

1

op

«delegate»
EvaluationDelegate

Double x
Double y = 0

Operator

- operatorInfo: Dictionary<String, Tuple<Int32, String, EvaluationDelegate>>
- op: Token

+ Op { get; } : Token
+ Precedence { get; } : Int32
+ Associativity { get; } : String
+ Operation { get; } : EvaluationDelegate
+ Operator(token: Token)