

Manual Técnico da Calculadora Polonesa Reversa

Arthur Bezerra Melo, Gabriel Bullé de Barros

16 de junho de 2024



Figure 1: Universidade de Pernambuco

1 Introdução e Visão Geral

Este manual descreve o software da Calculadora Polonesa Reversa implementada em C++. A versão atual é 1.0.0.

2 Implementação e Funcionamento

2.1 Classe **Stack**

A classe `Stack` implementa uma pilha genérica usando um vetor para armazenar os elementos. Ela fornece métodos para empilhar (`push`), desempilhar (`pop`), acessar o topo (`top`), verificar se está vazia (`isEmpty`) e obter o tamanho (`size`).

```

1  template <typename T>
2  class Stack
3  {
4      private:
5          std::vector<T> elements;
6
7      public:
8          void push(const T &element)
9          {
10             elements.push_back(element);
11         }
12         void pop()
13         {
14             if(elements.empty())
15                 throw std::out_of_range("Stack<>::pop(): pilha
16                     vazia");
17             elements.pop_back();
18         }
19         T top() const
20         {
21             if(elements.empty())
22                 throw std::out_of_range("Stack<>::top(): pilha
23                     vazia");
24             return elements.back();
25         }
26         bool isEmpty() const
27         {
28             return elements.empty();
29         }
30
31         size_t size() const
32         {
33             return elements.size();
34         }
35     };
36 };

```

Listing 1: Implementação da classe Stack

2.2 Classe PolishCalculator

A classe PolishCalculator avalia expressões matemáticas na notação polonesa reversa. Ela utiliza uma instância da classe Stack para manipular os operandos e operadores durante a avaliação.

```

1 // Implemento da calculadora polonesa
2 class PolishCalculator
3 {
4     public:
5         double evaluate(const std::string &expression)
6         {
7             std::istringstream tokens(expression);
8             std::string token;
9             Stack<double> stack;
10
11             while(tokens >> token)
12             {
13                 if(isOperator(token))
14                 {
15                     double b = stack.top();
16                     stack.pop();
17                     double a = stack.top();
18                     stack.pop();
19                     double result = applyOperator(token, a, b);
20                     std::cout << "Aplicando operador " << token
21                         << " a " << a << " e " << b << " resulta
22                         em " << result << std::endl;
23                     stack.push(result);
24                 }
25             }
26         }
27     };

```

Listing 2: Implementação da classe PolishCalculator

2.2.1 Método **evaluate**

O método `evaluate` recebe uma expressão em formato de string e a avalia passo a passo:

- Separa os tokens da expressão usando um `std::istringstream`.
- Para cada token:
 - Empilha números na pilha.
 - Quando encontra um operador, desempilha os dois operandos mais recentes, aplica o operador e empilha o resultado de volta.
- Ao final, verifica se há exatamente um valor na pilha. Se não, lança uma exceção indicando uma "Expressão inválida".

3 Função main

A função main controla a execução do programa, interagindo com o usuário e utilizando a calculadora polonesa reversa.

```
1 int main()
2 {
3     PolishCalculator calculator;
4     std::string expression;
5
6     std::cout << "Digite uma expressao em notacao polonesa
7     reversa: ";
8     std::getline(std::cin, expression); // Leitura da expressao
9     do usuario
10
11     try
12     {
13         double result = calculator.evaluate(expression);
14         std::cout << "Resultado: " << result << std::endl;
15     }
16     catch(const std::exception &ex)
17     {
18         std::cerr << "Error: " << ex.what() << std::endl;
19     }
20
21     return 0;
22 }
```

Listing 3: Implementação da função principal Main

4 Instruções de Compilação, Uso e Resolução de Problemas

4.1 Instruções de Compilação

Para compilar o programa, use o seguinte comando:

```
$make ex10.out
```

4.2 Instruções de Uso

Para usar o programa, execute o binário gerado e digite uma expressão em notação polonesa reversa quando solicitado.

Exemplo de entrada:

```
Digite uma expressão em notação polonesa reversa: 5 10 * 25 / 5 *
```

Saída esperada:

```
Resultado: 10
```

4.3 Resolução de Problemas

Ao usar a calculadora, esteja atento aos seguintes erros:

- **Erro: Expressão inválida - faltam operandos:** Indica que a expressão fornecida não possui operandos suficientes para aplicar um operador.
- **Erro: Divisão por zero:** Indica que houve uma tentativa de dividir por zero durante a avaliação da expressão.
- **Erro: Operador inválido:** Indica que foi encontrado um operador desconhecido ou inválido na expressão.

Para suporte adicional, entre em contato com [gbb1@poli.br, abm4@poli.br].