

Estruturas de Dados - Trabalho 1

Generated by Doxygen 1.8.13

Contents

1	Estruturas de Dados - Trabalho 1	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	elemento Struct Reference	7
4.1.1	Detailed Description	7
4.1.2	Member Data Documentation	7
4.1.2.1	proximo	7
4.1.2.2	valor	7
4.2	t_pilha Struct Reference	8
4.2.1	Detailed Description	8
4.2.2	Member Data Documentation	8
4.2.2.1	topo	8

5 File Documentation	9
5.1 inc/calculadora.h File Reference	9
5.1.1 Detailed Description	9
5.1.2 Function Documentation	9
5.1.2.1 calcular()	9
5.2 inc/conversao.h File Reference	10
5.2.1 Detailed Description	10
5.2.2 Function Documentation	10
5.2.2.1 avaliaExpressao()	10
5.2.2.2 infixToPostfix()	10
5.2.2.3 prioridade()	11
5.2.2.4 resolucao()	11
5.2.2.5 validaExpressao()	11
5.3 inc/pilha.h File Reference	12
5.3.1 Detailed Description	13
5.3.2 Typedef Documentation	13
5.3.2.1 t_elemento	13
5.3.3 Function Documentation	13
5.3.3.1 alocaElemento()	13
5.3.3.2 alocaPilha()	13
5.3.3.3 imprimirPilha()	13
5.3.3.4 liberaPilha()	14
5.3.3.5 pilhaVazia()	14
5.3.3.6 pop()	14
5.3.3.7 push()	15
Index	17

Chapter 1

Estruturas de Dados - Trabalho 1

Felipe de Oliveira Hargreaves - 15/0009313

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

elemento	Estrutura para elementos da pilha	7
t_pilha	Estrutura de inicialização da pilha	8

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

inc/ calculadora.h	
Calculadora para notação posfixa	9
inc/ conversao.h	
Avaliação de expressões	10
inc/ pilha.h	
Estrutura de Pilha	12

Chapter 4

Class Documentation

4.1 elemento Struct Reference

Estrutura para elementos da pilha.

```
#include <pilha.h>
```

Public Attributes

- double [valor](#)
- struct [elemento](#) * [proximo](#)

4.1.1 Detailed Description

Estrutura para elementos da pilha.

A estrutura é implementada com base em uma lista encadeada, de forma dinâmica.

4.1.2 Member Data Documentation

4.1.2.1 proximo

```
struct elemento* elemento::proximo
```

Ponteiro para o proximo elemento da pilha

4.1.2.2 valor

```
double elemento::valor
```

Valor a ser armazenado no elemento

The documentation for this struct was generated from the following file:

- inc/[pilha.h](#)

4.2 t_pilha Struct Reference

Estrutura de inicialização da pilha.

```
#include <pilha.h>
```

Public Attributes

- [t_elemento](#) * [topo](#)

4.2.1 Detailed Description

Estrutura de inicialização da pilha.

Contém um ponteiro para o elemento topo da estrutura, informação essencial em uma pilha.

4.2.2 Member Data Documentation

4.2.2.1 topo

```
t\_elemento* t_pilha::topo
```

Indica o elemento no topo da pilha.

The documentation for this struct was generated from the following file:

- [inc/pilha.h](#)

Chapter 5

File Documentation

5.1 inc/calculadora.h File Reference

Calculadora para notação posfixa.

```
#include "pilha.h"
```

Functions

- void `calcular` ()
Função de implementação da calculadora.

5.1.1 Detailed Description

Calculadora para notação posfixa.

5.1.2 Function Documentation

5.1.2.1 `calcular()`

```
void calcular ( )
```

Função de implementação da calculadora.

Aceita qualquer valor real positivo. Recebe valores e os empilha, exibindo o status da pilha a cada valor adicionado. Ao receber um operador, realiza a operação e exibe a pilha atualizada. Possui duas operações especiais, sendo estas de repetição de operação e cópia de operando. Continua realizando operações até o número de operandos na pilha ser insuficiente ou até o usuário selecionar a opção de saída. Limpa o buffer de entrada a cada valor inserido para evitar leituras inválidas.

5.2 inc/conversao.h File Reference

Avaliação de expressões.

```
#include "pilha.h"
```

Functions

- int [validaExpressao](#) (char *expressao)
Avalia uma expressão infixa recebida.
- int [prioridade](#) (char c)
Define prioridade entre operadores aritméticos e delimitadores.
- void [infixToPostfix](#) (char *expressao, char *expressao_pos)
Passa uma expressão infixa para sua forma posfixa.
- void [avaliaExpressao](#) (char *expressao_pos)
Resolve uma expressão no formato posfixo.
- void [resolucao](#) (char *expressao)
Processo completo de avaliação, conversão e resolução da expressão.

5.2.1 Detailed Description

Avaliação de expressões.

Métodos para avaliação, conversão e resolução de expressões posfixas.

5.2.2 Function Documentation

5.2.2.1 [avaliaExpressao\(\)](#)

```
void avaliaExpressao (
    char * expressao_pos )
```

Resolve uma expressão no formato posfixo.

A função recebe a string previamente convertida. Transforma os caracteres numéricos em valores do tipo double, com cuidado necessário para tratar números de vários dígitos. Ao detectar um operador realiza a operação com os dois últimos valores encontrados. Trata como erro a presença de qualquer caracter não-numérico que não seja um operador. Ao final, exibe o resultado da expressão.

Parameters

<i>expressao_pos</i>	String com a expressão no formato posfixo.
----------------------	--

5.2.2.2 [infixToPostfix\(\)](#)

```
void infixToPostfix (
```

```
char * expressao,  
char * expressao_pos )
```

Passa uma expressão infixa para sua forma posfixa.

A conversão gera uma expressão 100% equivalente com a original. Podem haver pequenas inconsistências de formatação (ausência ou excesso de espaços). Ao final, imprime a expressão convertida.

Parameters

<i>expressao</i>	A expressão original, já validada.
<i>expressao_pos</i>	Inicialmente uma string vazia, após a conversão contém a expressão posfixa.

5.2.2.3 prioridade()

```
int prioridade (  
    char c )
```

Define prioridade entre operadores aritméticos e delimitadores.

Função de apoio para a função de conversão infixa-posfixa.

Parameters

<i>c</i>	Qualquer um dos 4 operadores básicos ou delimitadores ({, {, {, {.
----------	--

Returns

A prioridade estabelecida do caracter.

5.2.2.4 resolucao()

```
void resolucao (  
    char * expressao )
```

Processo completo de avaliação, conversão e resolução da expressão.

Agrupar as funções já definidas em um único procedimento. Recebe apenas a expressão original e faz com ela todo o tratamento especificado.

Parameters

<i>expressao</i>	String com a expressão a ser manipulada e resolvida.
------------------	--

5.2.2.5 validaExpressao()

```
int validaExpressao (  
    char * expressao )
```

Avalia uma expressão infixa recebida.

Analisa os delimitadores de prioridade (parênteses, colchetes, chaves) em uma expressão aritmética e informa se a expressão é válida ou não. O critério é a correspondência entre inicializadores e finalizadores de escopo.

Parameters

<code>expressao</code>	Expressão a ser avaliada.
------------------------	---------------------------

Returns

1 caso válida, 0 caso contrário.

5.3 inc/pilha.h File Reference

Estrutura de Pilha.

Classes

- struct [elemento](#)
Estrutura para elementos da pilha.
- struct [t_pilha](#)
Estrutura de inicialização da pilha.

Typedefs

- typedef struct [elemento](#) [t_elemento](#)
Estrutura para elementos da pilha.

Functions

- [t_pilha](#) * [alocaPilha](#) ()
Inicializa uma nova pilha.
- void [liberaPilha](#) ([t_pilha](#) *pilha)
Remove uma pilha.
- int [pilhaVazia](#) ([t_pilha](#) *pilha)
Indica se existem ou não elementos na pilha.
- [t_elemento](#) * [alocaElemento](#) ()
Inicializa um novo elemento da pilha.
- void [push](#) ([t_pilha](#) *pilha, double valor)
Operação de push.
- double [pop](#) ([t_pilha](#) *pilha)
Operação de pop.
- void [imprimirPilha](#) ([t_pilha](#) *pilha)
Impressão da pilha.

5.3.1 Detailed Description

Estrutura de Pilha.

Estrutura pilha e suas funções de manipulação.

5.3.2 Typedef Documentation

5.3.2.1 t_elemento

```
typedef struct elemento t_elemento
```

Estrutura para elementos da pilha.

A estrutura é implementada com base em uma lista encadeada, de forma dinâmica.

5.3.3 Function Documentation

5.3.3.1 alocaElemento()

```
t_elemento* alocaElemento ( )
```

Inicializa um novo elemento da pilha.

Cria um elemento com valor e ponteiro nulos.

Returns

Ponteiro para o elemento alocado.

5.3.3.2 alocaPilha()

```
t_pilha* alocaPilha ( )
```

Inicializa uma nova pilha.

Cria uma pilha vazia, com `pilha->topo = NULL`.

Returns

Ponteiro para a pilha alocada.

5.3.3.3 imprimirPilha()

```
void imprimirPilha (
    t_pilha * pilha )
```

Impressão da pilha.

Imprime todos os elementos da pilha, a partir de seu topo. É exibido um índice para cada valor.

Parameters

<i>pilha</i>	Pilha a ser imprimida.
--------------	------------------------

5.3.3.4 liberaPilha()

```
void liberaPilha (  
    t_pilha * pilha )
```

Remove uma pilha.

Libera a memória de todos os elementos alocados na pilha, e então libera a própria pilha.

Parameters

<i>pilha</i>	Pilha a ser desalocada.
--------------	-------------------------

5.3.3.5 pilhaVazia()

```
int pilhaVazia (  
    t_pilha * pilha )
```

Indica se existem ou não elementos na pilha.

Returns

1 caso vazia, 0 caso contrário.

5.3.3.6 pop()

```
double pop (  
    t_pilha * pilha )
```

Operação de pop.

Desempilha um valor, removendo-o do topo da pilha. O valor anterior assume sua posição.

Parameters

<i>pilha</i>	Pilha a se remover o valor.
--------------	-----------------------------

Returns

O valor removido.

5.3.3.7 push()

```
void push (
    t_pilha * pilha,
    double valor )
```

Operação de push.

Empilha um valor, criando um novo elemento e o colocando no topo da pilha.

Parameters

<i>pilha</i>	Pilha a se inserir o elemento.
<i>valor</i>	Número a ser inserido na pilha.

Index

- alocaElemento
 - [pilha.h, 13](#)
- alocaPilha
 - [pilha.h, 13](#)
- avaliaExpressao
 - [conversao.h, 10](#)
- calculadora.h
 - [calcular, 9](#)
- calcular
 - [calculadora.h, 9](#)
- conversao.h
 - [avaliaExpressao, 10](#)
 - [infixToPostfix, 10](#)
 - [prioridade, 11](#)
 - [resolucao, 11](#)
 - [validaExpressao, 11](#)
- elemento, [7](#)
 - [proximo, 7](#)
 - [valor, 7](#)
- imprimirPilha
 - [pilha.h, 13](#)
- [inc/calculadora.h, 9](#)
- [inc/conversao.h, 10](#)
- [inc/pilha.h, 12](#)
- infixToPostfix
 - [conversao.h, 10](#)
- liberaPilha
 - [pilha.h, 14](#)
- pilha.h
 - [alocaElemento, 13](#)
 - [alocaPilha, 13](#)
 - [imprimirPilha, 13](#)
 - [liberaPilha, 14](#)
 - [pilhaVazia, 14](#)
 - [pop, 14](#)
 - [push, 14](#)
 - [t_elemento, 13](#)
- pilhaVazia
 - [pilha.h, 14](#)
- pop
 - [pilha.h, 14](#)
- prioridade
 - [conversao.h, 11](#)
- proximo
 - [elemento, 7](#)
- push
 - [pilha.h, 14](#)
- resolucao
 - [conversao.h, 11](#)
- t_elemento
 - [pilha.h, 13](#)
- t_pilha, [8](#)
 - [topo, 8](#)
- topo
 - [t_pilha, 8](#)
- validaExpressao
 - [conversao.h, 11](#)
- valor
 - [elemento, 7](#)