

My Project

Generated by Doxygen 1.8.13

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	celula Struct Reference	5
3.1.1	Detailed Description	5
3.1.2	Member Data Documentation	5
3.1.2.1	proximo	5
3.1.2.2	valor	6
3.2	TipoPilha Struct Reference	6
3.2.1	Detailed Description	6
3.2.2	Member Data Documentation	6
3.2.2.1	tamanho	6
3.2.2.2	tamanhomax	6
3.2.2.3	topo	6
4	File Documentation	7
4.1	pilha.h File Reference	7
4.2	pilhafunc.cpp File Reference	7
4.2.1	Function Documentation	7
4.2.1.1	criaPilha()	7
4.2.1.2	DestroyStack()	8
4.2.1.3	IsEmpty()	8
4.2.1.4	IsFull()	8
4.2.1.5	Pop()	8
4.2.1.6	Push()	9
4.2.1.7	SetSize()	9
4.2.1.8	Size()	9
4.2.1.9	Top()	10
	Index	11

Chapter 1

Class Index

1.1 Class List

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

celula	Definicao de uma celula da pilha	5
TipoPilha	Definicao da Pilha	6

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

pilha.h	7
pilhafunc.cpp	7

Chapter 3

Class Documentation

3.1 celula Struct Reference

Definicao de uma celula da pilha.

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

Collaboration diagram for celula:

Public Attributes

- [TipodaPilha valor](#)
- [apontadorPilha * proximo](#)

3.1.1 Detailed Description

Definicao de uma celula da pilha.

3.1.2 Member Data Documentation

3.1.2.1 proximo

```
apontadorPilha\* celula::proximo
```

3.1.2.2 valor

`TipodaPilha` `celula::valor`

Tipo que a pilha vai armazenar.

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

- [pilha.h](#)

3.2 TipoPilha Struct Reference

Definicao da Pilha.

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

Collaboration diagram for TipoPilha:

Public Attributes

- [apontadorPilha](#) * [topo](#)
- int [tamanho](#)
- unsigned int [tamanhomax](#)

3.2.1 Detailed Description

Definicao da Pilha.

3.2.2 Member Data Documentation

3.2.2.1 tamanho

```
int TipoPilha::tamanho
```

3.2.2.2 tamanhomax

```
unsigned int TipoPilha::tamanhomax
```

3.2.2.3 topo

```
apontadorPilha* TipoPilha::topo
```

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

- [pilha.h](#)

Chapter 4

File Documentation

4.1 pilha.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "catch.hpp"
Include dependency graph for pilha.h:
```

4.2 pilhafunc.cpp File Reference

```
#include "pilha.h"
Include dependency graph for pilhafunc.cpp:
```

Functions

- [TipoPilha](#) * [criaPilha](#) (void)
- int [Push](#) ([TipoPilha](#) *monte, [TipodaPilha](#) insercao)
- [apontadorPilha](#) * [Pop](#) ([TipoPilha](#) *monte)
- [apontadorPilha](#) * [Top](#) ([TipoPilha](#) *monte)
- int [Size](#) ([TipoPilha](#) *monte)
- void [SetSize](#) ([TipoPilha](#) *monte, unsigned int tamanho1)
- bool [IsFull](#) ([TipoPilha](#) *monte)
- bool [IsEmpty](#) ([TipoPilha](#) *monte)
- void [DestroyStack](#) ([TipoPilha](#) *monte)

4.2.1 Function Documentation

4.2.1.1 criaPilha()

```
TipoPilha* criaPilha (
    void )
```

A funcao [criaPilha](#), cria uma pilha, inicialmente vazia. Seta o tamanho inicial para 0 e o tamanho maximo para 100

4.2.1.2 DestroyStack()

```
void DestroyStack (
    TipoPilha * monte )
```

Destroi os elementos da pilha, fazendo com que vire uma Pilha vazia. Recebe como entrada a Pilha

4.2.1.3 IsEmpty()

```
bool IsEmpty (
    TipoPilha * monte )
```

Indica se a Pilha esta vazia. Recebe como entrada a Pilha

Returns

Retorna true caso a pilha esteja vazia

4.2.1.4 IsFull()

```
bool IsFull (
    TipoPilha * monte )
```

Indica se a pilha ja esta cheia. Recebe como entrada a Pilha

Returns

Retorna true caso a pilha ja esteja cheia, false caso contrario

4.2.1.5 Pop()

```
apontadorPilha* Pop (
    TipoPilha * monte )
```

Remove um elemento da Pilha e retorna esse mesmo elemento

Returns

Retorna a celula da pilha que foi removida, com todos seus dados

Parameters

<i>TipoPilha</i>	ponteiro(monte)
------------------	-----------------

4.2.1.6 Push()

```
int Push (
    TipoPilha * monte,
    TipodaPilha insercao )
```

Insere um elemento na Pilha, contendo um valor (que é do tipo pilha) o tamanho da pilha é aumentado em 1

Returns

Retorna 0 se a insercao foi feita ou -1 caso tenha falhado

Parameters

TipoPilha	ponteiro(monte), TipoPilha insercao
---------------------------	---

4.2.1.7 SetSize()

```
void SetSize (
    TipoPilha * monte,
    unsigned int tamanho1 )
```

Muda o tamanho maximo da pilha, inicialmente setado em 100 elementos Recebe como entrada a Pilha e o novo tamanho desejado

4.2.1.8 Size()

```
int Size (
    TipoPilha * monte )
```

Funcao que mostra o tamanho atual da pilha, por meio de uma das características da pilha (tamanho)

Returns

Retorna o tamanho da pilha como um inteiro

Parameters

TipoPilha	ponteiro(monte)
---------------------------	-----------------

4.2.1.9 Top()

```
apontadorPilha* Top (
    TipoPilha * monte )
```

Retorna o elemento que esta no topo da pilha, mas sem remove-lo

Returns

Retorna o elemento do topo da Pilha

Parameters

<i>TipoPilha</i>	ponteiro(monte)
------------------	-----------------

Index

- celula, [5](#)
 - proximo, [5](#)
 - valor, [5](#)
- criaPilha
 - pilhafunc.cpp, [7](#)
- DestroyStack
 - pilhafunc.cpp, [7](#)
- IsEmpty
 - pilhafunc.cpp, [8](#)
- IsFull
 - pilhafunc.cpp, [8](#)
- pilha.h, [7](#)
- pilhafunc.cpp, [7](#)
 - criaPilha, [7](#)
 - DestroyStack, [7](#)
 - IsEmpty, [8](#)
 - IsFull, [8](#)
 - Pop, [8](#)
 - Push, [9](#)
 - SetSize, [9](#)
 - Size, [9](#)
 - Top, [9](#)
- Pop
 - pilhafunc.cpp, [8](#)
- proximo
 - celula, [5](#)
- Push
 - pilhafunc.cpp, [9](#)
- SetSize
 - pilhafunc.cpp, [9](#)
- Size
 - pilhafunc.cpp, [9](#)
- tamanho
 - TipoPilha, [6](#)
- tamanhomax
 - TipoPilha, [6](#)
- TipoPilha, [6](#)
 - tamanho, [6](#)
 - tamanhomax, [6](#)
 - topo, [6](#)
- Top
 - pilhafunc.cpp, [9](#)
- topo
 - TipoPilha, [6](#)
- valor
 - celula, [5](#)