

Pilha_Encadeada

Generated by Doxygen 1.8.14

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	ElementoLista Struct Reference	5
3.1.1	Detailed Description	5
3.2	ItemType Struct Reference	5
3.2.1	Detailed Description	6
3.3	pilha Struct Reference	6
3.3.1	Detailed Description	6
4	File Documentation	7
4.1	include/pilha.h File Reference	7
4.1.1	Detailed Description	8
4.1.2	Function Documentation	8
4.1.2.1	create_stack()	8
4.1.2.2	empty()	9
4.1.2.3	exists()	9
4.1.2.4	free_stack()	9
4.1.2.5	isFULL()	10
4.1.2.6	pop()	10
4.1.2.7	print_pilha()	11

4.1.2.8	push()	11
4.1.2.9	set_size()	11
4.1.2.10	tamanho()	12
4.1.2.11	top()	12
4.2	Source/pilha.c File Reference	12
4.2.1	Detailed Description	13
4.2.2	Function Documentation	13
4.2.2.1	create_stack()	13
4.2.2.2	empty()	14
4.2.2.3	exists()	14
4.2.2.4	free_stack()	14
4.2.2.5	isFULL()	15
4.2.2.6	pop()	15
4.2.2.7	print_pilha()	16
4.2.2.8	push()	16
4.2.2.9	set_size()	16
4.2.2.10	tamanho()	17
4.2.2.11	top()	17
4.3	Source/teste_pilha.c File Reference	17
4.3.1	Detailed Description	18
Index		19

Chapter 1

Class Index

1.1 Class List

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

ElementoLista	Estrutura de um elemento da pilha	5
ItemType	Estrutura do tipo de dado utilizado no programa no caso Dado como ItemType	5
pilha	Estrutura da cabeça de uma pilha	6

Chapter 2

File Index

2.1 File List

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

include/ pilha.h	Arquivo de definição de funções	7
Source/ pilha.c	Arquivo de implementação das funções	12
Source/ teste_pilha.c	Arquivo de testes das funções	17

Chapter 3

Class Documentation

3.1 ElementoLista Struct Reference

Estrutura de um elemento da pilha.

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

Public Attributes

- [Dado](#) **pedaco**
- struct [ElementoLista](#) * **next**

3.1.1 Detailed Description

Estrutura de um elemento da pilha.

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

- include/[pilha.h](#)

3.2 ItemType Struct Reference

Estrutura do tipo de dado utilizado no programa no caso Dado como [ItemType](#).

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

Public Attributes

- int **x**

3.2.1 Detailed Description

Estrutura do tipo de dado utilizado no programa no caso Dado como [ItemType](#).

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

- [include/pilha.h](#)

3.3 pilha Struct Reference

Estrutura da cabeça de uma pilha.

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

Public Attributes

- [Elemento](#) * **inicio**
- int **size**
- int **max**

3.3.1 Detailed Description

Estrutura da cabeça de uma pilha.

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

- [include/pilha.h](#)

Chapter 4

File Documentation

4.1 include/pilha.h File Reference

Arquivo de definição de funções.

```
#include <stdlib.h>
#include <stdio.h>
```

Classes

- struct [ItemType](#)
Estrutura do tipo de dado utilizado no programa no caso Dado como [ItemType](#).
- struct [ElementoLista](#)
Estrutura de um elemento da pilha.
- struct [pilha](#)
Estrutura da cabeça de uma pilha.

Typedefs

- typedef struct [ItemType](#) [Dado](#)
Estrutura do tipo de dado utilizado no programa no caso Dado como [ItemType](#).
- typedef struct [ElementoLista](#) [Elemento](#)
Estrutura de um elemento da pilha.
- typedef struct [pilha](#) [Pilha](#)
Estrutura da cabeça de uma pilha.

Functions

- `Pilha * create_stack (Pilha *, int max_tam)`
Aloca memória para a pilha.
- `int push (Pilha *p, Dado elem)`
Adiciona um dado para a pilha.
- `int pop (Pilha *p)`
retira ultimo elemento da pilha
- `int empty (Pilha *p)`
Verifica se a pilha está vazia.
- `int print_pilha (Pilha *p)`
Printa toda a pilha.
- `Pilha * free_stack (Pilha *p)`
Apaga a pilha da memória.
- `int tamanho (Pilha *p)`
Retorna o tamanho atual da pilha.
- `int isFULL (Pilha *p)`
Verifica se a pilha atingiu o tamanho máximo.
- `int exists (Pilha *p)`
Verifica se a pilha foi alocada Nesse caso se ela existe.
- `int top (Pilha *p, Dado *elem)`
Retorna elemento do topo da pilha.
- `Pilha * set_size (Pilha *p, int tam)`
Redefine o tamanho da pilha.

4.1.1 Detailed Description

Arquivo de definição de funções.

Author

Luís Eduardo

Date

2018-09-13

4.1.2 Function Documentation

4.1.2.1 create_stack()

```
Pilha* create_stack (  
    Pilha * p,  
    int max_tam )
```

Aloca memória para a pilha.

Parameters

<i>p</i>	
<i>max_tam</i>	

Returns

Pilha*

4.1.2.2 empty()

```
int empty (
    Pilha * p )
```

Verifica se a pilha está vazia.

Parameters

<i>p</i>	
----------	--

Returns

int

4.1.2.3 exists()

```
int exists (
    Pilha * p )
```

Verifica se a pilha foi alocada Nesse caso se ela existe.

Parameters

<i>p</i>	
----------	--

Returns

int

4.1.2.4 free_stack()

```
Pilha* free_stack (
    Pilha * p )
```

Apaga a pilha da memória.

Parameters

p	
-----	--

Returns

Pilha*

4.1.2.5 isFULL()

```
int isFULL (
    Pilha * p )
```

Verifica se a pilha atingiu o tamanho máximo.

Parameters

p	
-----	--

Returns

int

4.1.2.6 pop()

```
int pop (
    Pilha * p )
```

retira ultimo elemento da pilha

Parameters

p	
-----	--

Returns

int

4.1.2.7 print_pilha()

```
int print_pilha (  
    Pilha * p )
```

Printa toda a pilha.

Parameters

<i>p</i>	
----------	--

Returns

int

4.1.2.8 push()

```
int push (  
    Pilha * p,  
    Dado elem )
```

Adiciona um dado para a pilha.

Parameters

<i>p</i>	
<i>elem</i>	

Returns

int

4.1.2.9 set_size()

```
Pilha* set_size (  
    Pilha * p,  
    int tam )
```

Redefine o tamanho da pilha.

Parameters

<i>p</i>	
<i>tam</i>	

Returns

Pilha*

4.1.2.10 tamanho()

```
int tamanho (
    Pilha * p )
```

Retorna o tamanho atual da pilha.

Parameters

<i>p</i>	
----------	--

Returns

int

4.1.2.11 top()

```
int top (
    Pilha * p,
    Dado * elem )
```

Retorna elemento do topo da pilha.

Parameters

<i>p</i>	
<i>elem</i>	

Returns

int

4.2 Source/pilha.c File Reference

Arquivo de implementação das funções.

```
#include <stdio.h>
#include <stdlib.h>
#include "../include/pilha.h"
```


Functions

- `Pilha * create_stack (Pilha *p, int max_tam)`
Aloca memória para a pilha.
- `int push (Pilha *p, Dado elem)`
Adiciona um dado para a pilha.
- `int top (Pilha *p, Dado *elem)`
Retorna elemento do topo da pilha.
- `int pop (Pilha *p)`
retira ultimo elemento da pilha
- `int empty (Pilha *p)`
Verifica se a pilha está vazia.
- `int print_pilha (Pilha *p)`
Printa toda a pilha.
- `Pilha * free_stack (Pilha *p)`
Apaga a pilha da memória.
- `int tamanho (Pilha *p)`
Retorna o tamanho atual da pilha.
- `int isFULL (Pilha *p)`
Verifica se a pilha atingiu o tamanho máximo.
- `int exists (Pilha *p)`
Verifica se a pilha foi alocada Nesse caso se ela existe.
- `Pilha * set_size (Pilha *p, int tam)`
Redefine o tamanho da pilha.

4.2.1 Detailed Description

Arquivo de implementação das funções.

Author

Luís Eduardo

Date

2018-09-13

4.2.2 Function Documentation

4.2.2.1 create_stack()

```
Pilha* create_stack (  
    Pilha * p,  
    int max_tam )
```

Aloca memória para a pilha.

Parameters

p	
max_tam	

Returns

Pilha*

4.2.2.2 empty()

```
int empty (
    Pilha * p )
```

Verifica se a pilha está vazia.

Parameters

p	
-----	--

Returns

int

4.2.2.3 exists()

```
int exists (
    Pilha * p )
```

Verifica se a pilha foi alocada Nesse caso se ela existe.

Parameters

p	
-----	--

Returns

int

4.2.2.4 free_stack()

```
Pilha* free_stack (
    Pilha * p )
```

Apaga a pilha da memória.

Parameters

p	
-----	--

Returns

Pilha*

4.2.2.5 isFULL()

```
int isFULL (
    Pilha * p )
```

Verifica se a pilha atingiu o tamanho máximo.

Parameters

p	
-----	--

Returns

int

4.2.2.6 pop()

```
int pop (
    Pilha * p )
```

retira ultimo elemento da pilha

Parameters

p	
-----	--

Returns

int

4.2.2.7 print_pilha()

```
int print_pilha (
    Pilha * p )
```

Printa toda a pilha.

Parameters

<i>p</i>	
----------	--

Returns

int

4.2.2.8 push()

```
int push (
    Pilha * p,
    Dado elem )
```

Adiciona um dado para a pilha.

Parameters

<i>p</i>	
<i>elem</i>	

Returns

int

4.2.2.9 set_size()

```
Pilha* set_size (
    Pilha * p,
    int tam )
```

Redefine o tamanho da pilha.

Parameters

<i>p</i>	
<i>tam</i>	

Returns

Pilha*

4.2.2.10 tamanho()

```
int tamanho (
    Pilha * p )
```

Retorna o tamanho atual da pilha.

Parameters

<i>p</i>	
----------	--

Returns

int

4.2.2.11 top()

```
int top (
    Pilha * p,
    Dado * elem )
```

Retorna elemento do topo da pilha.

Parameters

<i>p</i>	
<i>elem</i>	

Returns

int

4.3 Source/teste_pilha.c File Reference

Arquivo de testes das funções.

```
#include <iostream>
#include "../include/pilha.h"
#include <gtest/gtest.h>
```

Functions

- **TEST** (existence_condition, testar_existencia_da_pilha)
Constroi um novo objeto de Teste Testa condição de existência da pilha.
- **TEST** (existence_condition_all_functions, testar_existencia_todas_funcoes)
Constroi um novo objeto de Teste Testa condição de existência de todas as funções.
- **TEST** (create_stack_test, Criar_pilha)
Constroi um novo objeto de Teste Testa a criação da pilha.
- **TEST** (free_pilha_test, teste_desalocamento_da_pilha)
Constroi um novo objeto de Teste Testa função free_stack.
- **TEST** (push_pilha_test, Adicionar_elemento_na_pilha)
Constroi um novo objeto de Teste Testa função push.
- **TEST** (pop_pilha_test, retirar_elemento_na_pilha)
Constroi um novo objeto de Teste Testa função pop.
- **TEST** (empty_pilha_test, checar_pilha_vazia)
Constroi um novo objeto de Teste Testa função empty.
- **TEST** (isFULL_test, checar_pilha_esta_cheia)
Constroi um novo objeto de Teste Testa função is FULL.
- **TEST** (set_size_test, checar_se_tamanho_foi_modificado)
Constroi um novo objeto de Teste Testa função set_size.
- **TEST** (top_function_test, testar_top)
Constroi um novo objeto de Teste Testa função top.
- **TEST** (tamanho_function_test, testar_tamanho)
Constroi um novo objeto de Teste Testa função tamanho.
- **TEST** (print_function_test, testar_print)
Constroi um novo objeto de Teste Testa função print.
- int **main** (int argc, char **argv)

4.3.1 Detailed Description

Arquivo de testes das funções.

Author

Luís Eduardo

Date

2018-09-13

Index

- create_stack
 - [pilha.c, 13](#)
 - [pilha.h, 8](#)
- ElementoLista, [5](#)
- empty
 - [pilha.c, 14](#)
 - [pilha.h, 9](#)
- exists
 - [pilha.c, 14](#)
 - [pilha.h, 9](#)
- free_stack
 - [pilha.c, 14](#)
 - [pilha.h, 9](#)
- include/pilha.h, [7](#)
- isFULL
 - [pilha.c, 15](#)
 - [pilha.h, 10](#)
- ItemType, [5](#)
-
- [pilha, 6](#)
- [pilha.c](#)
 - [create_stack, 13](#)
 - [empty, 14](#)
 - [exists, 14](#)
 - [free_stack, 14](#)
 - [isFULL, 15](#)
 - [pop, 15](#)
 - [print_pilha, 15](#)
 - [push, 16](#)
 - [set_size, 16](#)
 - [tamanho, 17](#)
 - [top, 17](#)
- [pilha.h](#)
 - [create_stack, 8](#)
 - [empty, 9](#)
 - [exists, 9](#)
 - [free_stack, 9](#)
 - [isFULL, 10](#)
 - [pop, 10](#)
 - [print_pilha, 10](#)
 - [push, 11](#)
 - [set_size, 11](#)
 - [tamanho, 12](#)
 - [top, 12](#)
- pop
 - [pilha.c, 15](#)
 - [pilha.h, 10](#)
-
- print_pilha
 - [pilha.c, 15](#)
 - [pilha.h, 10](#)
- push
 - [pilha.c, 16](#)
 - [pilha.h, 11](#)
-
- set_size
 - [pilha.c, 16](#)
 - [pilha.h, 11](#)
- Source/pilha.c, [12](#)
- Source/teste_pilha.c, [17](#)
-
- tamanho
 - [pilha.c, 17](#)
 - [pilha.h, 12](#)
- top
 - [pilha.c, 17](#)
 - [pilha.h, 12](#)