

Pilha_vetor

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	ItemType Struct Reference	5
3.1.1	Detailed Description	5
3.2	pilha Struct Reference	5
3.2.1	Detailed Description	5
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() [1/2]	12
4.1.2.12	top() [2/2]	13
4.2	Source/pilha.c File Reference	13
4.2.1	Detailed Description	14
4.2.2	Function Documentation	14
4.2.2.1	create_stack()	14
4.2.2.2	empty()	14
4.2.2.3	exists()	15
4.2.2.4	free_stack()	15
4.2.2.5	isFULL()	15
4.2.2.6	pop()	16
4.2.2.7	print_pilha()	16
4.2.2.8	push()	17
4.2.2.9	set_size()	17
4.2.2.10	tamanho()	17
4.2.2.11	top()	18
4.3	Source/teste_pilha.c File Reference	18
4.3.1	Detailed Description	19
Index		21

Chapter 1

Class Index

1.1 Class List

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

ItemType	Tipo de elemento inserido na pilha	5
pilha	Estrutura básica da pilha	5

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	7
Source/ pilha.c	
Arquivo de funções	13
Source/ teste_pilha.c	
Arquivo de testes	18

Chapter 3

Class Documentation

3.1 ItemType Struct Reference

Tipo de elemento inserido na pilha.

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

Public Attributes

- int **x**

3.1.1 Detailed Description

Tipo de elemento inserido na pilha.

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

- [include/pilha.h](#)

3.2 pilha Struct Reference

Estrutura básica da pilha

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

Public Attributes

- [Dado](#) * **valor**
- int **max_tam**
- int **size**

3.2.1 Detailed Description

Estrutura básica da 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.

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

Classes

- struct [ItemType](#)
Tipo de elemento inserido na pilha.
- struct [pilha](#)
Estrutura básica da pilha

Typedefs

- typedef struct [ItemType](#) [Dado](#)
Tipo de elemento inserido na pilha.
- typedef struct [pilha](#) [Pilha](#)
Estrutura básica da pilha

Functions

- [Pilha *](#) [create_stack](#) ([Pilha](#) *p, int max_tam)
Criar estrutura básica da pilha com tamanho máximo definido.
- int [top](#) ([Pilha](#) *p, [Dado](#) element)
Pega primeiro elemento da pilha.
- int [push](#) ([Pilha](#) *p, [Dado](#) elem)
Adiciona elemento para a pilha.
- int [pop](#) ([Pilha](#) *p, [Dado](#) *elemento)

- Retira ultimo elemento da pilha.*
- int `empty` (`Pilha *p`)
Verifica se a pilha esta vazia.
- int `print_pilha` (`Pilha *p`)
printa todos os elementos da pilha
- `Pilha * free_stack` (`Pilha *p`)
limpa a pilha
- int `tamanho` (`Pilha *p`)
verifica tamanho da pilha
- int `isFULL` (`Pilha *p`)
verifica se a pilha atinge o tamanho maximo
- int `exists` (`Pilha *p`)
verifica se a pilha existe
- int `top` (`Pilha *p`, `Dado *elemento`)
Acessa ultimo elemento da pilha.
- `Pilha * set_size` (`Pilha *p`, int `tamanho`)
Redefine o tamanho maximo da pilha.

4.1.1 Detailed Description

Arquivo de definição.

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 )
```

Criar estrutura básica da pilha com tamanho máximo definido.

Parameters

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

Returns

Pilha*

4.1.2.2 empty()

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

Verifica se a pilha esta vazia.

Parameters

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

Returns

int

4.1.2.3 exists()

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

verifica se a pilha existe

Parameters

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

Returns

int

4.1.2.4 free_stack()

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

limpa a pilha

Parameters

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

Returns

Pilha*

4.1.2.5 isFULL()

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

verifica se a pilha atinge o tamanho maximo

Parameters

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

Returns

int

4.1.2.6 pop()

```
int pop (
    Pilha * p,
    Dado * elemento )
```

Retira ultimo elemento da pilha.

Parameters

<i>p</i>	
<i>elemento</i>	

Returns

int

4.1.2.7 print_pilha()

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

printa todos os elementos da pilha

Parameters

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

Returns

int

4.1.2.8 push()

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

Adiciona elemento 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 maximo da pilha.

Parameters

<i>p</i>	
<i>tamanho</i>	

Returns

Pilha*

Parameters

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

Returns

Pilha*

4.1.2.10 tamanho()

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

verifica tamanho da pilha

Parameters

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

Returns

int

4.1.2.11 top() [1/2]

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

Pega primeiro elemento da pilha.

Parameters

<i>p</i>	
<i>element</i>	

Returns

int

4.1.2.12 top() [2/2]

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

Acessa ultimo elemento da pilha.

Parameters

<i>p</i>	
<i>elemento</i>	

Returns

int

4.2 Source/pilha.c File Reference

Arquivo de funções.

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

Functions

- `Pilha * create_stack (Pilha *p, int max_tam)`
Criar estrutura básica da pilha com tamanho máximo definido.
- `int push (Pilha *p, Dado elem)`
Adiciona elemento para a pilha.
- `int pop (Pilha *p, Dado *elemento)`
Retira ultimo elemento da pilha.
- `int empty (Pilha *p)`
Verifica se a pilha esta vazia.
- `int print_pilha (Pilha *p)`
printa todos os elementos da pilha
- `Pilha * free_stack (Pilha *p)`
limpa a pilha
- `int tamanho (Pilha *p)`
verifica tamanho da pilha
- `int isFULL (Pilha *p)`
verifica se a pilha atinge o tamanho maximo
- `int exists (Pilha *p)`
verifica se a pilha existe
- `int top (Pilha *p, Dado *elemento)`
Acessa ultimo elemento da pilha.
- `Pilha * set_size (Pilha *p, int tam)`
Redefine o tamanho maximo da pilha.

4.2.1 Detailed Description

Arquivo de 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 )
```

Criar estrutura básica da pilha com tamanho máximo definido.

Parameters

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

Returns

Pilha*

4.2.2.2 empty()

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

Verifica se a pilha esta vazia.

Parameters

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

Returns

int

4.2.2.3 exists()

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

verifica se a pilha existe

Parameters

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

Returns

int

4.2.2.4 free_stack()

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

limpa a pilha

Parameters

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

Returns

Pilha*

4.2.2.5 isFULL()

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

verifica se a pilha atinge o tamanho maximo

Parameters

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

Returns

int

4.2.2.6 pop()

```
int pop (
    Pilha * p,
    Dado * elemento )
```

Retira ultimo elemento da pilha.

Parameters

<i>p</i>	
<i>elemento</i>	

Returns

int

4.2.2.7 print_pilha()

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

printa todos os elementos da pilha

Parameters

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

Returns

int

4.2.2.8 push()

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

Adiciona elemento 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 maximo da pilha.

Parameters

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

Returns

Pilha*

4.2.2.10 tamanho()

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

verifica tamanho da pilha

Parameters

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

Returns

int

4.2.2.11 top()

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

Acessa ultimo elemento da pilha.

Parameters

<i>p</i>	
<i>elemento</i>	

Returns

int

4.3 Source/teste_pilha.c File Reference

Arquivo de testes.

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

Functions

- **TEST** (Pilha_exists_tests, test_existences)
Constroi um novo objeto de Teste Teste da função exists.
- **TEST** (create_stack_test, Criar_pilha)
Constroi um novo objeto de Teste Teste da função create_stack.
- **TEST** (push_pilha_test, Adicionar_elemento_na_pilha)
Constroi um novo objeto de Teste Teste da função push_pilha.
- **TEST** (pop_pilha_test, retirar_elemento_na_pilha)
Constroi um novo objeto de Teste Teste da função pop.
- **TEST** (empty_pilha_test, checar_pilha_vazia)
Constroi um novo objeto de Teste Teste da função empty.
- **TEST** (isFULL_test, checar_pilha_esta_cheia)
Constroi um novo objeto de Teste Teste da função isFULL.
- **TEST** (Pilha_exists_test, checar_se_a_pilha_existe)
Constroi um novo objeto de Teste Teste da função exists 2.
- **TEST** (Pilha_top_access, checar_primeiro_elemento_da_pilha)

- Constroi um novo objeto de Teste Teste da função top.*
 - **TEST** (pilha_set_size_test, testar_alteracao_tamanho)
- Constroi um novo objeto de Teste Teste da função set_size.*
 - **TEST** (print_pilha_test, testar_print_da_pilha)
- Constroi um novo objeto de Teste Teste da função print.*
 - **TEST** (free_stack_test, testar_free_stack_function)
- Constroi um novo objeto de Teste Teste da função free_stack.*
 - **TEST** (tamanho_test, testar_tamanho_da_pilha)
- Constroi um novo objeto de Teste Teste da função tamanho.*
 - int **main** (int argc, char **argv)

4.3.1 Detailed Description

Arquivo de testes.

Author

Luís Eduardo

Date

2018-09-13

Index

- create_stack
 - [pilha.c, 14](#)
 - [pilha.h, 8](#)
- empty
 - [pilha.c, 14](#)
 - [pilha.h, 9](#)
- exists
 - [pilha.c, 15](#)
 - [pilha.h, 9](#)
- free_stack
 - [pilha.c, 15](#)
 - [pilha.h, 9](#)
- include/pilha.h, [7](#)
- isFULL
 - [pilha.c, 15](#)
 - [pilha.h, 10](#)
- ItemType, [5](#)
- [pilha, 5](#)
- [pilha.c](#)
 - [create_stack, 14](#)
 - [empty, 14](#)
 - [exists, 15](#)
 - [free_stack, 15](#)
 - [isFULL, 15](#)
 - [pop, 16](#)
 - [print_pilha, 16](#)
 - [push, 16](#)
 - [set_size, 17](#)
 - [tamanho, 17](#)
 - [top, 18](#)
- [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, 16](#)
 - [pilha.h, 10](#)
- [print_pilha](#)
 - [pilha.c, 16](#)
 - [pilha.h, 10](#)
- [push](#)
 - [pilha.c, 16](#)
 - [pilha.h, 11](#)
- [set_size](#)
 - [pilha.c, 17](#)
 - [pilha.h, 11](#)
- [Source/pilha.c, 13](#)
- [Source/teste_pilha.c, 18](#)
- [tamanho](#)
 - [pilha.c, 17](#)
 - [pilha.h, 12](#)
- [top](#)
 - [pilha.c, 18](#)
 - [pilha.h, 12](#)