

Estruturas de Dados na Linguagem C

Gerado por Doxygen 1.8.7

Domingo, 4 de Dezembro de 2016 12:32:47

Sumário

1	Índice das Estruturas de Dados	1
1.1	Estruturas de Dados	1
2	Índice dos Arquivos	3
2.1	Lista de Arquivos	3
3	Estruturas	5
3.1	Referência da Estrutura fila	5
3.1.1	Descrição Detalhada	5
3.1.2	Campos	5
3.1.2.1	frente	5
3.1.2.2	tras	5
3.2	Referência da Estrutura lista	5
3.2.1	Descrição Detalhada	6
3.2.2	Campos	6
3.2.2.1	primeiro	6
3.2.2.2	ultimo	6
3.3	Referência da Estrutura Noh	6
3.3.1	Descrição Detalhada	6
3.3.2	Campos	6
3.3.2.1	item	6
3.3.2.2	prox	6
3.4	Referência da Estrutura pilha	7
3.4.1	Descrição Detalhada	7
3.4.2	Campos	7
3.4.2.1	fundo	7
3.4.2.2	tamanho	7
3.4.2.3	topo	7
4	Arquivos	9
4.1	Referência do Arquivo Fila.c	9
4.1.1	Descrição Detalhada	9

4.1.2	Funções	9
4.1.2.1	ApagaFila	9
4.1.2.2	CriaFila	10
4.1.2.3	Desenfileira	10
4.1.2.4	Enfileira	10
4.1.2.5	Vazia	10
4.2	Referência do Arquivo Fila.h	11
4.2.1	Descrição Detalhada	11
4.2.2	Funções	11
4.2.2.1	ApagaFila	11
4.2.2.2	CriaFila	11
4.2.2.3	Desenfileira	12
4.2.2.4	Enfileira	12
4.2.2.5	Vazia	12
4.3	Referência do Arquivo Lista.c	12
4.3.1	Descrição Detalhada	13
4.3.2	Funções	13
4.3.2.1	CriaLista	13
4.3.2.2	Imprime	13
4.3.2.3	Insere	13
4.3.2.4	Retira	14
4.3.2.5	Vazia	14
4.4	Referência do Arquivo Lista.h	14
4.4.1	Descrição Detalhada	15
4.4.2	Funções	15
4.4.2.1	CriaLista	15
4.4.2.2	Imprime	15
4.4.2.3	Insere	15
4.4.2.4	Retira	15
4.4.2.5	Vazia	15
4.5	Referência do Arquivo Noh.h	16
4.5.1	Descrição Detalhada	16
4.6	Referência do Arquivo Pilha.c	16
4.6.1	Descrição Detalhada	17
4.6.2	Funções	17
4.6.2.1	ApagaPilha	17
4.6.2.2	CriaPilha	17
4.6.2.3	Desempilha	17
4.6.2.4	Empilha	17
4.6.2.5	Tamanho	18

4.6.2.6	Vazia	18
4.7	Referência do Arquivo Pilha.h	18
4.7.1	Descrição Detalhada	19
4.7.2	Funções	19
4.7.2.1	ApagaPilha	19
4.7.2.2	CriaPilha	19
4.7.2.3	Desempilha	19
4.7.2.4	Empilha	20
4.7.2.5	Tamanho	20
4.7.2.6	Vazia	20

Capítulo 1

Índice das Estruturas de Dados

1.1 Estruturas de Dados

Aqui estão as estruturas de dados, uniões e suas respectivas descrições:

fila	Definicao do TAD fila	5
lista	Definicao do TAD lista	5
Noh	Definicao do TAD noh	6
pilha	Definicao do TAD fila	7

Capítulo 2

Índice dos Arquivos

2.1 Lista de Arquivos

Esta é a lista de todos os arquivos documentados e suas respectivas descrições:

Fila.c	Conteúdo das funções referentes ao tipo abstrato de dado Fila	9
Fila.h	Protótipos das funções referentes ao tipo abstrato de dado Fila	11
Lista.c	Conteúdo das funções referentes ao tipo abstrato de dado Lista	12
Lista.h	Protótipos das funções referentes ao tipo abstrato de dado Lista	14
Noh.h	Conteúdo do tipo abstrato de dado Noh	16
Pilha.c	Conteúdo das funções referentes ao tipo abstrato de dado Pilha	16
Pilha.h	Protótipos das funções referentes ao tipo abstrato de dado Pilha	18

Capítulo 3

Estruturas

3.1 Referência da Estrutura fila

Definicao do TAD fila.

```
#include <Fila.h>
```

Diagrama de colaboração para fila:

Campos de Dados

- [noh * frente](#)
- [noh * tras](#)

3.1.1 Descrição Detalhada

Definicao do TAD fila.

3.1.2 Campos

3.1.2.1 [noh* fila::frente](#)

ponteiro para o primeiro elemento da fila, ou seja, a sua frente.

3.1.2.2 [noh* fila::tras](#)

ponteiro para o ultimo elemento da fila, ou seja, a sua traseira.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- [Fila.h](#)

3.2 Referência da Estrutura lista

Definicao do TAD lista.

```
#include <Lista.h>
```

Diagrama de colaboração para lista:

Campos de Dados

- `noh * primeiro`
- `noh * ultimo`

3.2.1 Descrição Detalhada

Definicao do TAD lista.

3.2.2 Campos

3.2.2.1 `noh* lista::primeiro`

ponteiro para o primeiro elemento da lista.

3.2.2.2 `noh* lista::ultimo`

ponteiro para o ultimo elemento da lista.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- [Lista.h](#)

3.3 Referência da Estrutura Noh

Definicao do TAD noh.

```
#include <Noh.h>
```

Diagrama de colaboração para Noh:

Campos de Dados

- `int item`
- `struct Noh * prox`

3.3.1 Descrição Detalhada

Definicao do TAD noh.

3.3.2 Campos

3.3.2.1 `int Noh::item`

item armazenado no noh

3.3.2.2 `struct Noh* Noh::prox`

ponteiro para o proximo noh.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- [Noh.h](#)

3.4 Referência da Estrutura pilha

Definicao do TAD fila.

```
#include <Pilha.h>
```

Diagrama de colaboração para pilha:

Campos de Dados

- `noh * fundo`
- `noh * topo`
- `int tamanho`

3.4.1 Descrição Detalhada

Definicao do TAD fila.

3.4.2 Campos

3.4.2.1 `noh* pilha::fundo`

ponteiro para o primeiro elemento empilhado, ou seja, o seu fundo.

3.4.2.2 `int pilha::tamanho`

total de itens empilhados.

3.4.2.3 `noh* pilha::topo`

ponteiro para o ultimo elemento empilhado, ou seja, o seu topo.

A documentação para esta estrutura foi gerada a partir do seguinte arquivo:

- [Pilha.h](#)

Capítulo 4

Arquivos

4.1 Referência do Arquivo Fila.c

Conteúdo das funções referentes ao tipo abstrato de dado Fila.

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

Gráfico de dependência de inclusões para Fila.c:

Funções

- void **CriaFila** (fila *f)
Cria uma fila vazia, atribuindo o endereço a um ponteiro passado como parametro.
- int **Vazia** (fila f)
Verifica se a fila passada como parametro esta vazia.
- void **Enfileira** (int x, fila *f)
Enfileira o item x na fila f.
- void **Desenfileira** (fila *f, int *item)
Desenfileira um item da fila armazenando seu valor na variavel item.
- void **ApagaFila** (fila *f)
Apaga toda a fila, inclusive o nó vazio localizado na frente da fila.

4.1.1 Descrição Detalhada

Conteúdo das funções referentes ao tipo abstrato de dado Fila.

Este arquivo contém os conteúdos das funções do tipo abstrato de dado Fila.

Autor

Frederico S. Oliveira

4.1.2 Funções

4.1.2.1 void ApagaFila (fila * f)

Apaga toda a fila, inclusive o nó vazio localizado na frente da fila.

Remove todos os elementos da fila, um por um, incluindo o primeiro nó vazio, desalocando o espaço de memória de cada item. Esta função tem objetivo apagar toda a memória alocada.

Parâmetros

<i>fila</i>	ponteiro para a fila.
-------------	-----------------------

4.1.2.2 void CriaFila (fila * f)

Cria uma fila vazia, atribuindo o endereço a um ponteiro passado como parametro.

O primeiro elemento de uma fila eh sempre um noh vazio. Ao criar uma fila vazia, aponta-se os ponteiros frente e tras para esse noh vazio.

Parâmetros

<i>fila</i>	ponteiro do tipo fila que receberah o endereço da fila criada.
-------------	--

4.1.2.3 void Desenfileira (fila * f, int * item)

Desenfileira um item da fila armazenando seu valor na variavel item.

Para desenfileirar o elemento da frente, deve-se lembrar o que primeiro elemento da lista eh um noh vazio. Portanto, o valor a ser desenfileirado encontra-se no segundo noh. Para isso, remove-se o primeiro elemento (noh vazio) e aponta o ponteiro frente para o segundo elemento na fila. Em seguida, copia o valor do noh desenfileirado para a variavel item. Por fim, desaloca o elemento que era o primeiro elemento da fila.

Parâmetros

<i>fila</i>	ponteiro para a fila da qual serah desenfileirado o item.
<i>item</i>	ponteiro que armazenarah o valor do item desenfileirado.

4.1.2.4 void Enfileira (int x, fila * f)

Enfileira o item x na fila f.

Para enfileirar um elemento aloca-se um novo noh. Aponta-se o campo prox do ultimo elemento da fila para o novo noh. Copia o valor de x para o novo noh. Aponta o campo prox do novo noh para NULL.

Parâmetros

<i>item</i>	a ser inserido na fila.
<i>fila</i>	ponteiro para a fila na qual serah enfileirado o novo item.

4.1.2.5 int Vazia (fila f)

Verifica se a fila passada como parametro estah vazia.

Para verificar se a fila estah vazia, realiza-se o teste se os ponteiros frente e tras apontam para o mesmo noh, o noh vazio inserido na funcao CriaFila.

Parâmetros

<i>fila</i>	variavel do tipo fila. Observe que nao eh um ponteiro.
-------------	--

Retorna

Zero (true) se fila vazia, um (false) se nao-vazia.

4.2 Referência do Arquivo Fila.h

Prototipos das funcoes referentes ao tipo abstrato de dado Fila.

```
#include "Noh.h"
```

Gráfico de dependência de inclusões para Fila.h: Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:

Estruturas de Dados

- struct [fila](#)

Definicao do TAD fila.

Funções

- void [CriaFila](#) ([fila](#) *f)

Cria uma fila vazia, atribuindo o endereco a um ponteiro passado como parametro.

- int [Vazia](#) ([fila](#) f)

Verifica se a fila passada como parametro estah vazia.

- void [Enfileira](#) (int x, [fila](#) *f)

Enfileira o item x na fila f.

- void [Desenfileira](#) ([fila](#) *f, int *item)

Desenfileira um item da fila armazenando seu valor na variavel item.

- void [ApagaFila](#) ([fila](#) *f)

Apaga toda a fila, inclusive o noh vazio localizado na frente da fila.

4.2.1 Descrição Detalhada

Prototipos das funcoes referentes ao tipo abstrato de dado Fila.

Este arquivo contem os prototipos das funcoes do tipo abstrato de dado Fila.

Autor

Frederico S. Oliveira

4.2.2 Funções

4.2.2.1 void ApagaFila ([fila](#) * f)

Apaga toda a fila, inclusive o noh vazio localizado na frente da fila.

Remove todos os elementos da fila, um por um, incluindo o primeiro noh vazio, desalocando o espaco de memoria de cada item. Esta funcao tem objetivo apagar todo a memoria alocada.

Parâmetros

fila	ponteiro para a fila.
----------------------	-----------------------

4.2.2.2 void CriaFila ([fila](#) * f)

Cria uma fila vazia, atribuindo o endereco a um ponteiro passado como parametro.

O primeiro elemento de uma fila eh sempre um noh vazio. Ao criar uma fila vazia, aponta-se os ponteiros frente e tras para esse noh vazio.

Parâmetros

<i>fila</i>	ponteiro do tipo fila que receberá o endereço da fila criada.
-------------	---

4.2.2.3 void Desenfileira (fila * f, int * item)

Desenfileira um item da fila armazenando seu valor na variável item.

Para desenfileirar o elemento da frente, deve-se lembrar o que primeiro elemento da lista é um nó vazio. Portanto, o valor a ser desenfileirado encontra-se no segundo nó. Para isso, remove-se o primeiro elemento (nó vazio) e aponta o ponteiro frente para o segundo elemento na fila. Em seguida, copia o valor do nó desenfileirado para a variável item. Por fim, desaloca o elemento que era o primeiro elemento da fila.

Parâmetros

<i>fila</i>	ponteiro para a fila da qual será desenfileirado o item.
<i>item</i>	ponteiro que armazenará o valor do item desenfileirado.

4.2.2.4 void Enfileira (int x, fila * f)

Enfileira o item x na fila f.

Para enfileirar um elemento aloca-se um novo nó. Aponta-se o campo prox do último elemento da fila para o novo nó. Copia o valor de x para o novo nó. Aponta o campo prox do novo nó para NULL.

Parâmetros

<i>item</i>	a ser inserido na fila.
<i>fila</i>	ponteiro para a fila na qual será enfileirado o novo item.

4.2.2.5 int Vazia (fila f)

Verifica se a fila passada como parâmetro está vazia.

Para verificar se a fila está vazia, realiza-se o teste se os ponteiros frente e trás apontam para o mesmo nó, o nó vazio inserido na função CriaFila.

Parâmetros

<i>fila</i>	variável do tipo fila. Observe que não é um ponteiro.
-------------	---

Retorna

Zero (true) se fila vazia, um (false) se não-vazia.

4.3 Referência do Arquivo Lista.c

Conteúdo das funções referentes ao tipo abstrato de dado Lista.

```
#include <sys/time.h>
#include <stdlib.h>
#include <stdio.h>
#include "Lista.h"
```

Gráfico de dependência de inclusões para Lista.c:

Funções

- void **CriaLista** (**lista** *)
Cria uma lista vazia, atribuindo o endereço a um ponteiro passado como parametro.
- int **Vazia** (**lista** l)
Verifica se a lista passada como parametro estah vazia.
- void **Inserere** (int x, **lista** *)
Inserere o item x no final da lista l.
- void **Retira** (**noh** *p, **lista** *, int *item)
Retira o item localizado apos o noh p na lista.
- void **Imprime** (**lista** l)
Imprime o conteudo de todos os nohs da lista.

4.3.1 Descrição Detalhada

Conteudo das funcoes referentes ao tipo abstrato de dado Lista.

Este arquivo contem os conteudos das funcoes do tipo abstrato de dado Lista.

Autor

Frederico S. Oliveira

4.3.2 Funções

4.3.2.1 void CriaLista (lista * l)

Cria uma lista vazia, atribuindo o endereço a um ponteiro passado como parametro.

Parâmetros

<i>lista</i>	ponteiro do tipo lista que receberah o endereço da lista criada.
--------------	--

4.3.2.2 void Imprime (lista l)

Imprime o conteudo de todos os nohs da lista.

Parâmetros

<i>lista</i>	ponteiro apontando para a lista que serah impressa.
--------------	---

4.3.2.3 void Inserere (int x, lista * l)

Inserere o item x no final da lista l.

Primeiramente aloca um novo noh. Adiciona esse novo noh no final da lista. Copia o valor do item para o novo noh. O novo noh passa a ser o ultimo noh da lista. Atualiza o campo ultimo da fila, fazendo-o apontar para o novo noh, que eh o ultimo.

Parâmetros

<i>item</i>	a ser inserido na lista.
-------------	--------------------------

<i>lista</i>	ponteiro do tipo lista apontando para a lista na qual serah inserido o novo item.
--------------	---

4.3.2.4 void Retira (noh * p, lista * l, int * item)

Retira o item localizado apos o noh p na lista.

Apona o ponteiro q para o noh a ser removido da lista. Copia o valor de q para outra variavel. Atualiza o noh p, alterando seu campo prox, pois devemos "pular" o noh q. Por fim, deve-se verificar se foi removido o ultimo elemento, atualizando a lista.

Parâmetros

<i>lista</i>	ponteiro do tipo lista apontando para a lista da qual serah retirado o item.
<i>item</i>	ponteiro que armazenarah o valor do item removido da lista.

4.3.2.5 int Vazia (lista l)

Verifica se a lista passada como parametro estah vazia.

Parâmetros

<i>lista</i>	variavel do tipo lista. Observe que nao eh um ponteiro.
--------------	---

Retorna

Zero (true) se lista vazia, um (false) se nao-vazia.

4.4 Referência do Arquivo Lista.h

Prototipos das funcoes referentes ao tipo abstrato de dado Lista.

```
#include "Noh.h"
```

Gráfico de dependência de inclusões para Lista.h: Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:

Estruturas de Dados

- struct [lista](#)

Definicao do TAD lista.

Funções

- void [CriaLista](#) ([lista](#) *l)

Cria uma lista vazia, atribuindo o endereco a um ponteiro passado como parametro.

- int [Vazia](#) ([lista](#) l)

Verifica se a lista passada como parametro estah vazia.

- void [Insere](#) (int x, [lista](#) *l)

Insere o item x no final da lista l.

- void [Retira](#) (noh *p, [lista](#) *l, int *item)

Retira o item localizado apos o noh p na lista.

- void [Imprime](#) ([lista](#) l)

Imprime o conteudo de todos os nohs da lista.

4.4.1 Descrição Detalhada

Prototipos das funcoes referentes ao tipo abstrato de dado Lista.

Este arquivo contem os prototipos das funcoes do tipo abstrato de dado Lista.

Autor

Frederico S. Oliveira

4.4.2 Funções

4.4.2.1 void CriaLista (lista * l)

Cria uma lista vazia, atribuindo o endereco a um ponteiro passado como parametro.

Parâmetros

<i>lista</i>	ponteiro do tipo lista que receberah o endereco da lista criada.
--------------	--

4.4.2.2 void Imprime (lista l)

Imprime o conteudo de todos os nohs da lista.

Parâmetros

<i>lista</i>	ponteiro apontando para a lista que serah impressa.
--------------	---

4.4.2.3 void Insere (int x, lista * l)

Insere o item x no final da lista l.

Primeiramente aloca um novo noh. Adiciona esse novo noh no final da lista. Copia o valor do item para o novo noh. O novo noh passa a ser o ultimo noh da lista. Atualiza o campo ultimo da fila, fazendo-o apontar para o novo noh, que eh o ultimo.

Parâmetros

<i>item</i>	a ser inserido na lista.
<i>lista</i>	ponteiro do tipo lista apontando para a lista na qual serah inserido o novo item.

4.4.2.4 void Retira (noh * p, lista * l, int * item)

Retira o item localizado apos o noh p na lista.

Aponta o ponteiro q para o noh a ser removido da lista. Copia o valor de q para outra variavel. Atualiza o noh p, alterando seu campo prox, pois devemos "pular" o noh q. Por fim, deve-se verificar se foi removido o ultimo elemento, atualizando a lista.

Parâmetros

<i>lista</i>	ponteiro do tipo lista apontando para a lista da qual serah retirado o item.
<i>item</i>	ponteiro que armazenarah o valor do item removido da lista.

4.4.2.5 int Vazia (lista l)

Verifica se a lista passada como parametro estah vazia.

Parâmetros

<i>lista</i>	variavel do tipo lista. Observe que nao eh um ponteiro.
--------------	---

Retorna

Zero (true) se lista vazia, um (false) se nao-vazia.

4.5 Referência do Arquivo Noh.h

Conteudo do tipo abstrato de dado [Noh](#).

Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:

Estruturas de Dados

- struct [Noh](#)
Definicao do TAD noh.

Definições de Tipos

- typedef struct [Noh](#) noh
Definicao do TAD noh.

4.5.1 Descrição Detalhada

Conteudo do tipo abstrato de dado [Noh](#).

Autor

Frederico S. Oliveira

4.6 Referência do Arquivo Pilha.c

Conteudo das funcoes referentes ao tipo abstrato de dado Pilha.

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

Gráfico de dependência de inclusões para Pilha.c:

Funções

- void [CriaPilha](#) ([pilha](#) *p)
Cria uma pilha vazia, atribuindo o endereco a um ponteiro passado como parametro.
- int [Vazia](#) ([pilha](#) p)
Verifica se a pilha passada como parametro estah vazia.
- void [Empilha](#) (int x, [pilha](#) *p)
Empilha o item x na pilha p.
- void [Desempilha](#) ([pilha](#) *p, int *item)
Desempilha um item da pilha armazenando seu valor na variavel item.
- int [Tamanho](#) ([pilha](#) p)

Retorna o tamanho da pilha.

- void **ApagaPilha** (**pilha** *p)

Apaga toda a pilha, inclusive o noh vazio localizado no topo da pilha.

4.6.1 Descrição Detalhada

Conteúdo das funções referentes ao tipo abstrato de dado Pilha.

Este arquivo contém os conteúdos das funções do tipo abstrato de dado Pilha.

Autor

Frederico S. Oliveira

4.6.2 Funções

4.6.2.1 void **ApagaPilha** (**pilha** * p)

Apaga toda a pilha, inclusive o noh vazio localizado no topo da pilha.

Remove todos os elementos da pilha, um por um, incluindo o primeiro noh vazio, desalocando o espaço de memória de cada item. Esta função tem objetivo apagar toda a memória alocada.

Parâmetros

<i>pilha</i>	ponteiro para a pilha.
--------------	------------------------

4.6.2.2 void **CriaPilha** (**pilha** * p)

Cria uma pilha vazia, atribuindo o endereço a um ponteiro passado como parâmetro.

O elemento localizado no topo é sempre um noh vazio. Ao criar uma pilha vazia, aponta-se os ponteiros topo e fundo para esse noh vazio.

Parâmetros

<i>pilha</i>	ponteiro do tipo pilha que receberá o endereço da pilha criada.
--------------	---

4.6.2.3 void **Desempilha** (**pilha** * p, int * *item*)

Desempilha um item da pilha armazenando seu valor na variável item.

Para desempilhar o elemento do topo, deve-se lembrar que elemento no topo é sempre um noh vazio. Portanto, o valor a ser desempilhado encontra-se no segundo noh. Para isso, remove-se o primeiro elemento (noh vazio) e aponta o ponteiro topo para o segundo elemento na pilha. Em seguida, copia o valor do noh desempilhado para a variável item. Por fim, desaloca o elemento que era o elemento no topo da pilha.

Parâmetros

<i>pilha</i>	ponteiro para a pilha da qual será desempilhado o item.
<i>item</i>	ponteiro que armazenará o valor do item desempilhado.

4.6.2.4 void **Empilha** (int x, **pilha** * p)

Empilha o item x na pilha p.

Para empilhar um elemento na pilha deve-se lembrar que o elemento no topo é sempre um noh vazio. Portanto, o valor a ser empilhado é copiado para o noh no topo, que é vazio. Em seguida, deve-se inserir um novo noh no

topo. Para isso aloca-se um novo noh, apontando o campo prox desse novo noh para o topo. Por fim, aponta-se o ponteiro topo para o novo noh inserido.

Parâmetros

<i>item</i>	a ser inserido na pilha.
<i>pilha</i>	ponteiro para a pilha na qual serah enfileirado o novo item.

4.6.2.5 int Tamanho (*pilha p*)

Retorna o tamanho da pilha.

Parâmetros

<i>pilha</i>	ponteiro para a pilha.
--------------	------------------------

Retorna

quantidade de itens na pilha.

4.6.2.6 int Vazia (*pilha p*)

Verifica se a pilha passada como parametro estah vazia.

Para verificar se a pilha estah vazia, realiza-se o teste se os ponteiros topo e fundo apontam para o mesmo noh, o noh vazio inserido na funcao CriaPilha.

Parâmetros

<i>pilha</i>	variavel do tipo pilha. Observe que nao eh um ponteiro.
--------------	---

Retorna

Zero (true) se pilha vazia, um (false) se nao-vazia.

4.7 Referência do Arquivo Pilha.h

Prototipos das funcoes referentes ao tipo abstrato de dado Pilha.

```
#include "Noh.h"
```

Gráfico de dependência de inclusões para Pilha.h: Este grafo mostra quais arquivos estão direta ou indiretamente relacionados com este arquivo:

Estruturas de Dados

- struct [pilha](#)
Definicao do TAD fila.

Funções

- void [CriaPilha](#) ([pilha](#) *p)
Cria uma pilha vazia, atribuindo o endereco a um ponteiro passado como parametro.
- int [Vazia](#) ([pilha](#) p)
Verifica se a pilha passada como parametro estah vazia.
- void [Empilha](#) (int x, [pilha](#) *p)

- *Empilha o item x na pilha p.*
• void **Desempilha** (**pilha** *p, int *item)
Desempilha um item da pilha armazenando seu valor na variavel item.
- int **Tamanho** (**pilha** p)
Retorna o tamanho da pilha.
- void **ApagaPilha** (**pilha** *p)
Apaga toda a pilha, inclusive o noh vazio localizado no topo da pilha.

4.7.1 Descrição Detalhada

Prototipos das funcoes referentes ao tipo abstrato de dado Pilha.

Este arquivo contem os prototipos das funcoes do tipo abstrato de dado Pilha.

Autor

Frederico S. Oliveira

4.7.2 Funções

4.7.2.1 void ApagaPilha (**pilha** * p)

Apaga toda a pilha, inclusive o noh vazio localizado no topo da pilha.

Remove todos os elementos da pilha, um por um, incluindo o primeiro noh vazio, desalocando o espaco de memoria de cada item. Esta funcao tem objetivo apagar todo a memoria alocada.

Parâmetros

pilha	ponteiro para a pilha.
--------------	------------------------

4.7.2.2 void CriaPilha (**pilha** * p)

Cria uma pilha vazia, atribuindo o endereco a um ponteiro passado como parametro.

O elemento localizado no topo eh sempre um noh vazio. Ao criar uma pilha vazia, aponta-se os ponteiros topo e fundo para esse noh vazio.

Parâmetros

pilha	ponteiro do tipo pilha que receberah o endereco da pilha criada.
--------------	--

4.7.2.3 void Desempilha (**pilha** * p, int * item)

Desempilha um item da pilha armazenando seu valor na variavel item.

Para desempilhar o elemento do topo, deve-se lembrar que elemento no topo eh sempre um noh vazio. Portanto, o valor a ser desempilhado encontra-se no segundo noh. Para isso, remove-se o primeiro elemento (noh vazio) e aponta o ponteiro topo para o segundo elemento na pilha. Em seguida, copia o valor do noh desempilhado para a variavel item. Por fim, desaloca o elemento que era o elemento no topo da pilha.

Parâmetros

pilha	ponteiro para a pilha da qual serah desempilhado o item.
--------------	--

<i>item</i>	ponteiro que armazenará o valor do item desempilhado.
-------------	---

4.7.2.4 void Empilha (int x, pilha * p)

Empilha o item x na pilha p.

Para empilhar um elemento na pilha deve-se lembrar que o elemento no topo é sempre um nó vazio. Portanto, o valor a ser empilhado é copiado para o nó no topo, que é vazio. Em seguida, deve-se inserir um novo nó no topo. Para isso aloca-se um novo nó, apontando o campo prox desse novo nó para o topo. Por fim, aponta-se o ponteiro topo para o novo nó inserido.

Parâmetros

<i>item</i>	a ser inserido na pilha.
<i>pilha</i>	ponteiro para a pilha na qual será enfileirado o novo item.

4.7.2.5 int Tamanho (pilha p)

Retorna o tamanho da pilha.

Parâmetros

<i>pilha</i>	ponteiro para a pilha.
--------------	------------------------

Retorna

quantidade de itens na pilha.

4.7.2.6 int Vazia (pilha p)

Verifica se a pilha passada como parâmetro está vazia.

Para verificar se a pilha está vazia, realiza-se o teste se os ponteiros topo e fundo apontam para o mesmo nó, o nó vazio inserido na função CriaPilha.

Parâmetros

<i>pilha</i>	variável do tipo pilha. Observe que não é um ponteiro.
--------------	--

Retorna

Zero (true) se pilha vazia, um (false) se não-vazia.

Índice Remissivo

- fila, [5](#)
 - frente, [5](#)
 - tras, [5](#)
- frente
 - fila, [5](#)
- fundo
 - pilha, [7](#)
- item
 - Noh, [6](#)
- lista, [5](#)
 - primeiro, [6](#)
 - ultimo, [6](#)
- Noh, [6](#)
 - item, [6](#)
 - prox, [6](#)
- pilha, [7](#)
 - fundo, [7](#)
 - tamanho, [7](#)
 - topo, [7](#)
- primeiro
 - lista, [6](#)
- prox
 - Noh, [6](#)
- tamanho
 - pilha, [7](#)
- topo
 - pilha, [7](#)
- tras
 - fila, [5](#)
- ultimo
 - lista, [6](#)