

Giullio Emmanuel da Cruz Di Gerolamo

RA: 790965

Frequência F2

Arquivo Pilha.h

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <conio.h>
#define tam 10
struct pilha//base da pilha
{
    int num_elementos[tam];
    int topo;
};

//funcao para empilhar um elemento na pilha
void empilhar(struct pilha *pilha, int elem, bool *check );

//funcao para desempilhar um elemento da pilha
void desempilhar(struct pilha *pilha, int *elem, bool *check);

//funcao para criar a pilha
void criar(struct pilha *pilha);

//funcao para checar se a pilha esta vazia
bool vazia(struct pilha *pilha);

//funcao para checar se a pilha esta cheia
bool cheia(struct pilha *pilha);
```

Arquivo Pilha.cpp

```
#include "Pilha.h"

void empilhar(struct pilha *pilha, int elem, bool *check){
    int j = tam - 1;//variavel da posicao do topo da pilha (tamanho menos um)
    if(pilha->topo == j){//verifica se existe espaco para empilhar
        *check = false;
    }
    else{
        pilha->topo++;//aumenta a posicao do topo
        pilha->num_elementos[pilha->topo] = elem;//add o elemento novo no topo
        *check = true;
    }
}

void desempilhar(struct pilha *pilha, int *elem, bool *check){
    if(pilha->topo < 0){//verifica se existe algum elemento para desempilhar
        *check = false;
    }
    else{
        *elem = pilha->num_elementos[pilha->topo];//retira o elemento e armazena na variavel
        pilha->num_elementos[pilha->topo] = 0;//elimina o topo
        pilha->topo--;//diminui uma posicao no topo
        *check = false;
    }
}

void criar(struct pilha *pilha){//criando uma pilha com o topo em -1, significa que esta vazia
    pilha->topo = -1;
}

bool vazia(struct pilha *pilha){
    if(pilha->topo > 0){//verifica se o topo eh menor que 0 (pilha vazia)
        return true;
    }
    else{
        return false;
    }
}

bool cheia(struct pilha *pilha){
    int k = tam - 1;//variavel da posicao do topo da pilha (tamanho menos um)
    if(pilha->topo == k){//verifica se o topo atingiu o tamanho da pilha(pilha cheia)
        return true;
    }
    else{
        return false;
    }
}
```

Uma possível aplicação para uma Pilha

Uma comum aplicação de uma pilha no Sistema é o desfazer/refazer de editors de texto. Estes por sua vez armazenam os dados que foram apagados anteriormente em uma pilha para caso necessario refazer.

Arquivo main.cpp

Não consegui executar de forma correta o arquivo main, não consegui imprimir as informações da pilha nem mesmo o topo.

Seguem abaixo fotos das tentativas, porém o console simplesmente não apresentava nenhum resultado.

```
24 int main(int argc, char** argv) {
25
26     struct pilha *X;
27     for(int k = 0; k < tam; k++){
28         X->num_elementos[k] = 2;
29     }
30     for(int i = 0; i < tam; i++){
31         printf ("%d\\n", X->num_elementos[i]);
32     }
33
34 }
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdbool.h>
4 #include <conio.h>
5 #define tam 10
6
7 struct pilha//base da pilha
8 {
9     int num_elementos[tam];
10     int topo;
11     void empilhar(struct pilha *pilha, int elem, bool *check);
12
13     //funcao para desempilhar um elemento da pilha
14     void desempilhar(struct pilha *pilha, int *elem, bool *check);
15
16     //funcao para criar a pilha
17     void criar(struct pilha *pilha);
18
19     //funcao para checar se a pilha esta vazia
20     bool vazia(struct pilha *pilha);
21
22     //funcao para checar se a pilha esta cheia
23     bool cheia(struct pilha *pilha);
24 };
25
26 //funcao para empilhar um elemento na pilha
27 void empilhar(struct pilha *pilha, int elem, bool *check);
28
29 //funcao para desempilhar um elemento da pilha
30 void desempilhar(struct pilha *pilha, int *elem, bool *check);
31
32 //funcao para criar a pilha
33 void criar(struct pilha *pilha);
34
35 //funcao para checar se a pilha esta vazia
36 bool vazia(struct pilha *pilha);
37
38 //funcao para checar se a pilha esta cheia
39 bool cheia(struct pilha *pilha);
40
41
42
```

```

1 using namespace std;
2 #include <conio.h>
3 #include <stdio.h>
4 #include <iostream>
5 #include "Pilha.h"
6
7 /* run this program using the console pauser or add your own getch, system("pause") or input loop */
8
9 void imprimir(struct pilha *pilha){
10     for(int i = 0; i < tam; i++){
11         cout<< pilha->num_elementos[i];
12     }
13 }
14
15 void aleatorio(struct pilha *pilha){
16     for(int k = 0; k < tam; k++){
17         pilha->num_elementos[k] = rand() % 100 + 1;
18     }
19 }
20
21 int main(int argc, char** argv) {
22     struct pilha *X;
23     aleatorio(X);
24     imprimir(X);
25     return 0;
26 }
27
28
29
30
31
32
33

```

I

```

1 #include "Pilha.h"
2
3 void empilhar(struct pilha *pilha, int elem, bool *check){
4     int j = tam - 1; //variavel da posicao do topo da pilha (tamanho menos um)
5     if(pilha->topo == j){ //verifica se existe espaco para empilhar
6         *check = false;
7     }
8     else{
9         pilha->topo++; //aumenta a posicao do topo
10        pilha->num_elementos[pilha->topo] = elem; //add o elemento novo no topo
11        *check = true;
12    }
13 }
14
15 void desempilhar(struct pilha *pilha, int *elem, bool *check){
16     if(pilha->topo < 0){ //verifica se existe algum elemento para desempilhar
17         *check = false;
18     }
19     else{
20         *elem = pilha->num_elementos[pilha->topo]; //retira o elemento e armazena na variavel
21         pilha->num_elementos[pilha->topo] = 0; //elimina o topo
22         pilha->topo--; //diminui uma posicao no topo
23         *check = false;
24     }
25 }
26
27 void criar(struct pilha *pilha){ //criando uma pilha com o topo em -1, significa que esta vazia
28     pilha->topo = -1;
29 }
30
31 bool vazia(struct pilha *pilha){
32     if(pilha->topo > 0){ //verifica se o topo eh menor que 0 (pilha vazia)
33         return true;
34     }
35     else{
36         return false;
37     }
38 }
39
40 bool cheia(struct pilha *pilha){
41     int k = tam - 1; //variavel da posicao do topo da pilha (tamanho menos um)
42     if(pilha->topo == k){ //verifica se o topo atingiu o tamanho da pilha (pilha cheia)
43         return true;
44     }
45     else{
46         return false;
47     }
48 }
49

```

```
Project1 [Project1.exe] - [C:\Users\Administrator\Desktop\work\AED1\FI\Project1.exe]
Process exited after 1.073 seconds with return value 3221225477
Press any key to continue . . .

20      X->num_elementos[k] = rand() % 100 + 1;
21    }
22  }
23  */
24  int main(int argc, char** argv) {
25
26      struct pilha *X;
27
28      /*struct pilha p;
29      bool ok;
30      char op = 't';
31      int valor;
32      while (op != 's') {
33          cout << "digite: (e)empilhar, (d)desempilhar, (s)sair [enter]" << endl;
34          cin >> op;
35          switch (op) {
36              case 'e': cout << "digite valor INTEIRO para empilhar empilhar [enter]" << endl;
37                      cin >> valor; // CUIDADO: DIGITE UM INTEIRO MESMO!!
38                      pilha.empilhar(p, valor, ok);
39                      if (ok==true) cout << "> valor empilhado" << endl;
40                      else cout << "> nao conseguiu empilhar" << endl;
41                      break;
42                      case 'd': p.Desempilha(valor,ok);
43                      if (ok==true) cout << "valor desempilhado" << valor << endl;
44                      else cout << "nao conseguiu desempilhar" << endl;
45                      break;
46                      default: cout << "saindo... " << endl; op = 's'; break;
47                      }; // case
48                      imprime(p);
49                      } // while
50      */
51      for(int k = 0; k < tam; k++){
52          X->num_elementos[k] = 2;
53      }
54      for(int i = 0; i < tam; i++){
55          printf ("%d\n", X->num_elementos[i]);
56      }
57  }
58
59      break;
60      case 'd': p.Desempilha(valor,ok);
61      if (ok==true) cout << "valor desempilhado" << valor << endl;
62      else cout << "nao conseguiu desempilhar" << endl;
63      break;
64      default: cout << "saindo... " << endl; op = 's'; break;
65      }; // case
66      imprime(p);
67      } // while
68  */
69  for(int i = 0; i < tam; i++){
70      cout << X->num_elementos[i];
71  }
72  }
```

Compiler | Resources | Compile Log | Debug | Find Results | Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Administrator\Desktop\work\AED1\FI\Project1.exe
- Output Size: 1,835285167676 MiB
- Compilation Time: 0.04s

Line: 26 Col: 15 Sel: 0 Lines: 56 Length: 1231 Insert Done parsing in 0.125 seconds

Desculpe decepcionar /: