
Exercício: Pilha

SCC0202 ALGORITMOS E ESTRUTURAS DE DADOS I
23 DE AGOSTO DE 2018

MATHEUS DE SOUZA SERENO
Nº USP: 9368491

Exercício

pilha.h

```
1  #define tampilha 100
2  typedef float elem;
3  typedef struct pilha{
4      elem vet[tampilha];
5      int topo;
6  }Pilha;
7  void cria_pilha(Pilha* p);
8  int push(Pilha* p, elem X);
9  int pop(Pilha* p, elem* X);
10 elem top(Pilha* p, int* erro);
11 int isEmpty(Pilha* p);
12 void Empty(Pilha* p);
13 int show(Pilha* p);
```

pilha.c

```
1  #include "pilha.h"
2  #include <stdio.h>
3
4  void cria_pilha(Pilha* p){
5      p->topo = -1;
6      return;
7  }
8
9  int push(Pilha* p, elem X){
10     if(p->topo == tampilha-1)
11         return 1;
12     p->topo++;
13     p->vet[p->topo] = X;
14     return 0;
15 }
16
17 int pop(Pilha* p, elem* X){
18     if(isEmpty(p))
19         return 1;
20     *X = p->vet[p->topo];
21     p->topo--;
22     return 0;
23 }
24
25 elem top(Pilha* p, int* erro){
26     if(isEmpty(p)){
27         *erro = 1;
28         return -1;
29     }
```

```

30     *erro = 0;
31     return p->vet[p->topo];
32 }
33
34 int isEmpty(Pilha* p){
35     return (p->topo == -1);
36 }
37
38 void Empty(Pilha* p){
39     p->topo = -1;
40     return;
41 }
42
43 int show(Pilha* p){
44     if(isEmpty(p))
45         return 1;
46     printf("****Pilha****\n");
47     for(int i = p->topo; i>=0;i--)
48         printf("%.2f\n",p->vet[i] );
49     return 0;
50 }

```

teste.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "pilha.h"
4
5  void menu(Pilha* p){
6      cria_pilha(p);
7      int escolha,erro;
8      float resposta;
9      printf("BEM-VINDOS À SUA PILHA!!!\n\nEscolha uma das opções abaixo\n\n");
10     do{
11         printf("(1)Empilhar um elemento\n(2)Desempilhar um elemento\n");
12         printf("(3)Mostrar o elemento do topo\n(4)Verificar se a pilha está vazia\n");
13         printf("(5)Esvaziar pilha\n(6)Mostrar pilha\n(7)Sair\nOpção: ");
14         scanf(" %d",&escolha);
15         switch (escolha) {
16             case 1:
17                 printf("Qual o valor a ser empilhado\? \n");
18                 scanf(" %f",&resposta );
19                 if(push(p,resposta))
20                     printf("A pilha já está cheia.\n");
21                 else
22                     printf("Sucesso ao empilhar!\n" );
23                 break;
24             case 2:
25                 if(pop(p,&resposta))

```

```

26     printf("A pilha está vazia.\n");
27     else
28         printf("O elemento desempilhado foi %f\n",resposta );
29     break;
30     case 3:
31         resposta = top(p,&erro);
32         if(erro == 1)
33             printf("A pilha está vazia.\n");
34         else
35             printf("O elemento no topo da pilha é %f \n",resposta);
36         break;
37     case 4:
38         if(isEmpty(p))
39             printf("A pilha está vazia.\n");
40         else
41             printf("A pilha não  está vazia.\n");
42         break;
43     case 5:
44         Empty(p);
45         printf("Sucesso ao esvaziar!\n" );
46         break;
47     case 6:
48         if(show(p))
49             printf("A pilha está vazia.\n");
50         break;
51     case 7:
52         break;
53     default:
54         printf("Opção inválida!\n");
55         break;
56     }
57 }while(escolha!=7);
58 return;
59 }
60
61 int main(int argc, char* argv[]){
62     Pilha p;
63     menu(&p);
64     return 0;
65 }

```
