



Transcrição de aula

Disciplina	Programação Imperativa - 1º ano - LEI		
Secretário	Número: 49421	Nome: José David Alves	
Data:	2011-03-21	Nº Página	
Turno:	T1 - 9h-10h	Nº Alunos	

SUMÁRIO

O TAD stack = modelo de dados, principais funcionalidades, construção de um módulo usando structs e arrays.

LIFO - Last In First Out

Stack - SP - Stack Pointer (Primeira Posição da Stack)

- Empty; Top; Push; Pop

Modelos em C

- Interface

- ficheiros .h (headers)

- modelo de dados

- assinatura das funções externas

- Implementação

- ficheiros .c

- include do respetivo .h

- códigos das funções internas e externas



Transcrição (folha de continuação)

```
Stack.h
#ifndef -STACK
#define -STACK
[
:
#endif
]
Gecuda conteu
multiplos
indices, para mod
dramen varias vezes STACK.h

#define MAXSTACK 100
typedef struct sStack
{
int sp;
int stack [MAXSTACK];
} Stack;

Definir dados booleanos:
bool.h
#ifndef -bool
#define -bool
typedef int boolean;
#define TRUE 1
#define FALSE 0
#endif

#include "bool.h" (após para indicar que
esta na directory de
trabalho)

boolean empty (Stack s);
int top (Stack s);
Stack push (Stack s, int elem);
Stack pop (Stack s);
```



Transcrição (folha de continuação)

Stack.c

```
#include "stack.h"
boolean empty (stack s)
{ if (sp == 0) return true; } (s.sp == 0) ? TRUE : FALSE
else return false;
}
int top (stack s)
{ return s.stack[s.sp-1]; }
Stack push (stack s, int e)
{
  s.stack[s.sp] = e;
  s.sp++;
  return s;
}
Stack pop (stack s)
{
  s.sp--;
  return s;
}
```

Prog.c

```
#include "stack.h"
int main ()
{
  Stack s1, s2 = {2, {7, 13}};
  s1.sp = 0;
  s1 = push (s1, 20);
  s1 = push (push (s1, 33), 4);
}
```



Transcrição (folha de continuação)

Cria executavel
- % gcc -o prog.exe prog.c stack.c