Estruturas primitivas

* Cheia( )
  + bool DeuCerto recebe true se a pilha estiver cheia ->
* Vazia( )
  + bool DeuCerto recebe true se a pilha estiver vazia ->
* Empilha(Pilha x, obj y, bool DeuCerto )
  + coloca o objeto y no topo da pilha x, incrementa topo e DeuCerto = true, ( verificar se não está cheia antes)
* Desempilha(Pilha x, obj y, bool DeuCerto)
  + desempilha o topo e armazena em y, decrementa topo, deuCerto recebe true ( não desempilha se estiver Vazia )
* Cria(Pilha p )
  + inicializa a pilha p vazia ( topo == -1, elementos ==0 quando aplicável)

-------------------------------------------------------------------------------------------------------------

//

public Cria(Pilha \*p){

int i;

for(i=0;i<tamanho, i++)

p.elementos[i] = 0;

p.topo = -1;

}

//

public Cheia(Pilha \*p){

/\* if(p.elementos[tamanho - 1] != 0){

return true \*/

if(p.topo == tamanho - 1)

return true;

}

else{return false; }

}

//

public Vazia(Pilha \*p){

if(p.topo == -1)

return true;

else

return false;

}

//

public Empilha(Pilha \*p, char x, boolean DeuCerto){

if(Cheia(p) == false){

p.topo += 1;

p.elementos[p.topo] = x;

DeuCerto = true;

}

else{DeuCerto = false; }

}

//

Desempilha(Pilha \*p. char x, boolean DeuCerto){

if(Vazia(p) == false){

x = p.elementos[p.topo];

p.elementos[p.topo] = 0;

p.topo -= 1;

DeuCerto = true;

} else { DeuCerto = false; }

}

Pilha Burra vs Pilha inteligente

Verifica ou não as entradas

Arquitetura de software - definir

--------------------------------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------------------------------

var elemento

boolean MaisElementos(P1, P2){

Pilha pAux, pAux2;

TransfereElementos(P1, pAux);

TransfereElementos(P2, pAux2);

while(Vazia(pAux) == false && Vazia(pAux2) == false){

bool ok = true;

Desempilha(pAux,elemento,ok);

Desempilha(pAux2,elemento,ok);

}

if(Vazia(pAux) && !Vazia(pAux2){

print("A segunda pilha tem mais elementos);

} else if(!Vazia(pAux) && Vazia(pAux2){

print("A primeira pilha tem mais elementos);

}

}

-----------------------------------------------------

boolean algumIgualaX( x ){

}

boolean iguais( P1, P2) {

boolean diferentes;

Pilha paux, paux2;

TransfereElementos(P1, paux);

TransfereElementos(P2, paux2);

elemento1; elemento2;

while(Vazia(paux) == false && Vazia(paux2) == false){

desempilha(paux,elemento1,ok);

desempilha(paux2, elemento2, ok);

if(elemento1 != elemento2)

print("São Diferentes");

diferentes = true;

if(diferentes == false)

print("São Iguais");

}

}

Copiar pilha burra feita em C no livro e colar aqui e entender

Fazer Exerc 2.5 pag 42

Fazer exerc 2.6