Universidade Federal Fluminense

Matéria Estrutura de Dados

Professor Dalessandro Soares

Alunos: Alessandro Sampaio e Camila Alves

#include <stdio.h>

#include <stdlib.h>

#include "Fila.h"

#include "Filapri.h"

int main(void){

Filap f=NULL;

f=inicia(f);

int i=0,x=0,op=0,d=0;

char n[81];

char \*nome;

while (x!=5){

system ("cls");

printf("[ -- Digite a opcao desejada -- ]\n\n");

printf("1 - Inserir elemento\n");

printf("2 - Remover um elemento\n");

printf("3 - Imprimir\n");

printf("4 - Inicializar\n");

printf("5 - Sair\n");

scanf("%d",&x);

system ("cls");

switch (x){

case 1:{

printf("Digite o nome\n");

scanf(" %80[^\n]",n);

printf("\n");

printf("Digite a idade\n");

scanf("%d",&d);

if(d>=60){

inserirpri(f,n,d,'p');

printf("Elemento inserido com sucesso\n");

}

else{

inserirpri(f,n,d,'c');

printf("Elemento inserido com sucesso\n");

}

system("pause");

break;

}

case 2:{

removerpri(f,n,&d);

printf ("Nome: %s\n",n);

printf ("Idade: %d\n",d);

system("pause");

break;

}

case 3:{

printf("Digite a opcao desejada:\n\n");

printf("1 - Imprimir Fila prioritária\n");

printf("2 - Imprimir Fila Comun\n");

printf("3 - Imprimir Todos\n");

scanf("%d",&op);

system ("cls");

switch (op){

case 1:{

printf ("Fila Prioridade:\n\n");

imprimirpri(f,'p');

break;

}

case 2:{

printf ("Fila Comun:\n\n");

imprimirpri(f,'c');

break;

}

case 3:{

printf ("Fila Prioritaria e Comum:\n\n");

printf ("Fila Prioridade:\n");

imprimirpri(f,'p');

printf ("Fila Comum:\n");

imprimirpri(f,'c');

break;

}

}

system("pause");

break;

}

case 4:{

f=inicia(f);

printf ("Sua fila foi iniciada\n\n");

system("pause");

break;

}

case 5:{

f=destruirpri(f);

printf ("Fila encerrada\n\n");

system("pause");

break;

}

}

}

return 0;

}

//Filapri.h

#include "Fila.h"

#include <stdlib.h>

#include <stdio.h>

typedef struct filap\* Filap;

struct filap\* inicia(struct filap\* a);

void removerpri(struct filap\* f,char \*nome, int \*id);

void imprimirpri(struct filap\*f, char tipo);

struct filap\* destruirpri(struct filap\* f);

//Filapri.c

#include "Filapri.h"

struct filap{

Fila comun;

Fila pri;

};

struct filap\* inicia(struct filap\* a){

if(a!=NULL){

a->comun=inicializar(a->comun);

a->pri=inicializar(a->pri);

return a;

}

a=(struct filap\*)malloc(sizeof(struct filap));

a->comun=NULL;

a->pri=NULL;

return a;

}

void inserirpri(struct filap\*a, char \*nome, int id, char onde){

if(onde=='p')

a->pri=inserir(a->pri,nome,id);

else

a->comun=inserir(a->comun,nome,id);

}

void removerpri(struct filap\* f,char \*nome, int \*id){

char temp[81];

int tempid;

if(f->pri!=NULL)

f->pri=remover(f->pri, temp, &tempid);

else

f->comun=remover(f->comun,temp,&tempid);

strcpy(nome, temp);

(\*id)=tempid;

}

void imprimirpri(struct filap\*f, char tipo){

if(tipo=='p')

imprimir(f->pri);

else if(tipo=='c')

imprimir(f->comun);

else{

printf ("Fila Prioridades:\n");

imprimir(f->pri);

printf ("Fila Comum:\n");

imprimir(f->comun);

}

}

struct filap\* destruirpri(struct filap\* f){

f->pri=destruir(f->pri);

f->comun=destruir(f->comun);

free(f);

return NULL;

}

//Fila.h

typedef struct fila \*Fila;

Fila inserir(Fila f, char \*nome, int id);

Fila remover (Fila f,char \*nome,int \*id);

void imprimir (Fila f);

Fila inicializar (Fila f);

Fila destruir (Fila f);

//Fila.c

#include "Fila.h"

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

struct fila{

char nome[81];

int id;

struct fila \*prox;

};

Fila inserir(Fila f, char \*nome, int id){

if (f == NULL){

f=(Fila)malloc(sizeof(struct fila));

strcpy(f->nome,nome);

f->id=id;

f->prox=NULL;

return f;

}

else{

f->prox=inserir(f->prox,nome,id);

}

return f;

}

Fila remover(Fila f, char\* nome, int \*id){

if(f!=NULL){

Fila n=f->prox;

strcpy(nome, f->nome);

(\*id)=f->id;

free(f);

return n;

}

else{

printf("Lista vazia\n\n");

return f;

}

}

void imprimir(Fila f){

Fila p;

for(p=f;p!=NULL;p=p->prox){

printf("Nome: %s\n", p->nome);

printf("Idade: %d\n----------------------\n",p->id);

}

}

Fila inicializar (Fila f){

if(f!=NULL){

return f;

}

f=NULL;

return f;

}

Fila destruir(Fila f){

Fila t=f;

while(f!=NULL){

t=f->prox;

free(f);

f=t;

}

return f;

}