**CRUD**

#include <stdio.h> //para printf e algumas outras funcoes

#include <stdlib.h> ///para exit

//struct

typedef struct {

int NumOrdem;

char marca[20];

char tipo[20];

char cor[20];

float tamponta;

} canetas;

//Prototypes

int inserir();

void listar();

void atualizar();

void excluir();

void gravar();

void sair();

//defini o tamanho do array para 50, para limitar quantidade de canetas a ser armazenada, evitando que o programa tente armazenar mais canetas do que

//a memoria disponivel permite, o que poderia causar erros ou falhas.

canetas canetas\_contador[50];

//acompanha quantas canetas foram inseridas no array,

int contador = 0;//tambem incrementa cada caneta adicionada e saber quantas canetas estao armazenadas atualmente

int NumOrdem = 0;

// funcao principal

int main() {

int valor = 0; // inicializa valor

do {

printf("\n");

printf("\t\t\t\t\t\t\t EMPRESA CANETAS \t\t\t\t\t\t\t");

printf("\nDigite uma das opcoes dos numeros abaixo:\n");

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

printf("2-Listar produtos\n");

printf("3-Atualizar produtos\n");

printf("4-Excluir produtos\n");

printf("5-Gravar produto\n");

printf("6-Sair\n");

scanf("%d", &valor);

switch (valor) {

case 1:

inserir(canetas\_contador, &contador, &NumOrdem);//usando um  ponteiro para pegar a referencia do endereco

break;

case 2:

listar(canetas\_contador, &contador);

break;

case 3:

atualizar(canetas\_contador, &contador, &NumOrdem);

break;

case 4:

excluir(canetas\_contador, &contador);

break;

case 5:

gravar(canetas\_contador, &contador, &NumOrdem);

break;

case 6:

sair();

break;

default:

printf("Valor invalido! Tente novamente.\n");

break;

}

} while (valor != 6);

return 0;

}

// funcao inserir

int inserir(canetas canetas\_contador[], int \*contador, int \*NumOrdem) {//usando ponteiro para chamar o valor do contador

if (\*contador < 50) {

printf("\n");

printf("Digite a marca da caneta: ");

scanf("%19s", canetas\_contador[\*contador].marca);

        //limitar o tamanho da string lida da entrada

printf("Digite o tipo da caneta(Gel, Esferografica, Hidrografica, Apagavel): ");

scanf("%19s", canetas\_contador[\*contador].tipo);

printf("Digite a cor: ");

scanf("%19s", canetas\_contador[\*contador].cor);

printf("Digite o tamanho da ponta: ");

scanf("%f", &canetas\_contador[\*contador].tamponta);

        (\*contador)++; // Incrementa o contador aqui antes adicionar a caneta

canetas\_contador[\*contador].NumOrdem = ++(\*NumOrdem);

} else {

printf("\nEstoque cheio!!!\n");

}

return 0;

}

//funcao listar

void listar(canetas canetas\_contador[], int \*contador) {

canetas func1;

int i;

for(i = 0; i < \*contador; i++) {

printf("\n");

printf("Numero de Ordem: %d\n", canetas\_contador[i].NumOrdem);

printf("Nome: %s\n", canetas\_contador[i].marca);

printf("Tipo: %s\n", canetas\_contador[i].tipo);

printf("Cor: %s\n", canetas\_contador[i].cor);

printf("Tamanho da ponta: %.1f\n", canetas\_contador[i].tamponta);

printf("\n");

}

}

//funcao atualizar

void atualizar(canetas canetas\_contador[], int \*contador, int \*NumOrdem) {

int x;

printf("\n");

printf("DIGITE O INDICE DA CANETA QUE DESEJA ATUALIZAR (0 a %d): ", \*contador - 1);

scanf("%d", &x);

if (x >= 0 && x < \*contador) {

printf("DIGITE UMA NOVA MARCA:  ");

scanf("%19s", canetas\_contador[x].marca);

printf("DIGITE UM NOVO TIPO(GEL, ESFEROGRAFICA, HIDROGRAFICA, APAGAVEL): ");

scanf("%19s", canetas\_contador[x].tipo);

printf("DIGITE UMA NOVA COR: ");

scanf("%19s", canetas\_contador[x].cor);

printf("DIGITE O NOVO TAMANHO DE PONTA: ");

scanf("%f", &canetas\_contador[x].tamponta);

printf("Caneta atualizada com sucesso!\n");

} else {

printf("Indice invalido!\n");

}

}

//funcao excluir

void excluir(canetas canetas\_contador[], int \*contador) {

int x;

printf("\n");

printf("DIGITE O INDICE DA CANETA QUE DESEJA EXCLUIR (0 a %d): ", \*contador - 1);

scanf("%d", &x);

if (x >= 0 && x < \*contador) {

// Desloca os elementos subsequentes para preencher o espaco vazio

for (int i = x; i < \*contador - 1; i++) {

canetas\_contador[i] = canetas\_contador[i + 1];

}

(\*contador)--; // Decrementa o contador de canetas

printf("\n");

printf("Caneta excluida com sucesso!\n");

} else {

printf("indice invalido!\n");

}

}

//funcao gravar

void gravar(canetas canetas\_contador[], int \*contador, int \*NumOrdem) {

FILE \*file = fopen("canetas.txt", "w");

if (file == NULL) {//retorna nulo

printf("Erro ao abrir o arquivo!\n");

return;

}

for (int i = 0; i < \*contador; i++) {

fprintf(file, "%d %s %s %s %.1f\n", canetas\_contador[i].NumOrdem, canetas\_contador[i].marca, canetas\_contador[i].tipo, canetas\_contador[i].cor, canetas\_contador[i].tamponta);

}

fclose(file);

printf("Dados gravados com sucesso!\n");

}

//funcao sair

void sair(void) {

printf("\n");

printf("PROGRAMA ENCERRADO!!!");

exit(0);//encerra o programa

}