



## Aula de Exercícios de Prog. II – Módulos e Encapsulamento

1. Seu objetivo é criar um conjunto de funções para implementar um acervo de filmes, implementado como um vetor de ponteiros para DVDs.

Crie um projeto no Visual Studio e insira no seu projeto um arquivo “acervo.h” com o seguinte conteúdo:

```
typedef struct dvd Dvd;  
  
void acervo_inicializa(int n, Dvd** acervo);  
int acervo_cadastra(int n, Dvd** acervo, int i);  
int acervo_remove(int n, Dvd** acervo, int i);  
void acervo_imprime(int n, Dvd** acervo);  
int acervo_aluga(int n, Dvd** acervo, int cod);
```

Em seguida, crie uma função “acervo.c” que implementa a estrutura para o tipo Dvd e implementa as funções declaradas no arquivo “acervo.h”. Seu arquivo “acervo.c” deve ter a seguinte estrutura:

```
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include "acervo.h"  
  
struct dvd {  
    char titulo[91]; /* titulo do filme */  
    int codigo; /* codigo interno para identificar o filme */  
    char genero[21]; /* genero do filme, ex: drama, acao, comedia */  
    int alugueis; /* numero de vezes que o filme foi alugado */  
};  
  
void acervo_inicializa(int n, Dvd** acervo)  
{  
}  
  
void acervo_cadastra_dvd(int n, Dvd** acervo, int i)  
{  
}  
  
void acervo_remove_dvd(int n, Dvd** acervo, int i)  
{  
}  
  
void acervo_imprime(int n, Dvd** acervo)  
{  
}  
  
int acervo_aluga(int n, Dvd** acervo, int cod)  
{  
}
```

O comportamento esperado para as suas funções é o seguinte:





**acervo\_inicializa:** recebe um vetor de ponteiros para Dvd e o número máximo de elementos no vetor e coloca todos os elementos do vetor com valor NULL.

**acervo\_cadastra:** recebe um vetor de ponteiros para Dvd e o número máximo de elementos no vetor, assim como um índice *i*, e cadastra um novo DVD na posição *i*. A função deve ler os dados a serem cadastrados do teclado. Se já houver um Dvd no índice *i*, seus dados devem ser sobrescritos. Se não houver um elemento, a função deve alocar o espaço de memória para um novo Dvd. A função deve retornar 0 se o índice for inválido e 1 caso contrário.

**acervo\_remove:** recebe um vetor de ponteiros para Dvd e o número máximo de elementos no vetor, assim como um índice *i*, e remove o elemento da posição *i* do vetor. Sua função deve liberar a memória e colocar a posição com valor NULL. Se o índice *i* for inválido a função deve retornar 0. Se o vetor já tiver com valor NULL na posição *i*, a função deve retornar -1. Caso contrário a função deve retornar 1.

**acervo\_imprime:** recebe um vetor de ponteiros para Dvd e o número máximo de elementos no vetor, e imprime os dados de cada DVD na tela.

**acervo\_aluga:** recebe um vetor de ponteiros para Dvd e o número máximo de elementos no vetor, assim como um código de Dvd, e incrementa o número de aluguéis do Dvd correspondente ao código de 1. A função deve retornar 0 se não encontrar um Dvd com o código correspondente e 1 caso contrário.

Crie um arquivo separado para conter a função main de teste do seu programa. Sua função deverá fazer include para o arquivo "acervo.h".

Parabéns, você acaba de criar um Tipo Abstrato de Dados (TAD) acervo de Dvds!

