



Estruturas de Dados e Programação

UFSM00272

Professor Jonas Bulegon Gassen
jonas.gassen@ufsm.br

TAD

TIPO ABSTRATO DE DADOS

- O que é abstração nesse contexto?
- O código feito para array que aumenta de tamanho quando necessário, depende de algumas premissas, quais?

- Aquele código só funciona SE o usuário daquele array (programador), utilizar a função para adicionar elementos.
- Se o programador utilizar diretamente o vetor do registro, por exemplo, o tamanho não irá aumentar quando houver necessidade.
- Como resolvemos isso?



TAD

- Podemos utilizar um TAD!
- Assim, podemos esconder (abstrair) do usuário como o tipo foi implementado.
- Evitamos utilização inadequada;
- Desacopla implementação e utilização;
- Facilita reuso e manutenção;

- Esse TAD permitirá apenas algumas operações.
- Vamos chamar o TAD de VetorDinamico, o mesmo permitirá as seguintes operações:
 - Criar o vetor dinâmico
 - Adicionar um item no final do vetor
 - Remover um item em uma dada posição
 - Liberar a memória do mesmo
 - Imprimir todos itens do vetor

- Para alcançar estes objetivos, o programa deve ser dividido em vários arquivos. Teremos, basicamente:
 - Arquivos fonte “.c”
 - Arquivos de cabeçalho “.h”

- Para o nosso exemplo, teríamos:
- Um arquivo “.h” para definir o registro e os protótipos das funções
- Um arquivo “.c” para a implementação das funções
- Um arquivo “.c” para utilizar o tipo, onde estará o main.


```
#ifndef VETORDINAMICO_H

#define VETORDINAMICO_H

typedef struct {

    int tamanho, capacidade;

    int *elementos;

} VetorDinamico;

VetorDinamico* criar_vetor(int capacidade_inicial);

void liberar_vetor(VetorDinamico *vetor);

void adicionar_elemento(VetorDinamico *vetor, int elemento);

void remover_elemento(VetorDinamico *vetor, int posicao);

void imprimir_vetor(VetorDinamico *vetor);

int buscar_elemento(VetorDinamico *vetor, int valor);

#endif /* VETORDINAMICO_H */
```

- Em seguida, precisamos implementar cada uma das funções com protótipo definido no arquivo de cabeçalho.
- Podemos criar um arquivos “vetordinamico.c” para tal.
- Por fim, precisamos utilizar o TAD no “main.c”.
- Há exemplo no moodle com algumas funções em aberto para completar.

- Complete as duas funções pendentes;
- Modifique o código para adicionar uma nova função ao TAD, remova todas ocorrências de um determinado valor no vetor.
- Podem reutilizar as funções existentes, se acharem adequado.