



Estruturas de Dados

Tipos Abstratos de Dados

Fábio Duncan de Souza

Instituto Federal Fluminense

Sumário

Estruturas de
Dados

Fábio Duncan

Conceitos
Básicos

Vantagens de
Uso

Operações
Básicas

Exemplo - O
tipo File

Tipo Opaco

Exemplo de
Imple-
mentação

Pilhas

Filas

Referências

- 1 Conceitos Básicos
- 2 Vantagens de Uso
- 3 Operações Básicas
- 4 Exemplo - O tipo File
- 5 Tipo Opaco
- 6 Exemplo de Implementação
- 7 Pilhas
- 8 Filas



- Uma estrutura de dados é uma forma de armazenar e organizar os dados de modo que estes possam ser usados de forma eficiente.
- Alguns exemplos das estruturas de dados presentes na linguagem C são os arrays e structs, criadas a partir de tipos de dados básicos.



Tipos Abstratos de Dados

Fábio Duncan

Vantagens de Uso

Operações Básicas

Exemplo - O tipo File

Tipo Opaco

Exemplo de Imple- mentação

Pilhas

Filas

Referências

- Um tipo abstrato de dados, ou TAD, é um conjunto de dados estruturados e as operações que podem ser executadas sobre esses dados
 - Usados quando:
 - Os tipos de dados e as estruturas de dados presentes na linguagem não são suficientes para uma aplicação;
 - Necessita-se de uma melhor estruturação dos dados, assim como especificar quais operações estarão disponíveis para manipular esses dados;

Conceitos Básicos

Tipos Abstratos de Dados

Estruturas de Dados

Fábio Duncan

Vantagens de Uso

Operações Básicas

Exemplo - O tipo File

Tipo Opaco

Exemplo de Imple- mentação

Pilhas

Filas

Referências

- Basicamente, o tipo abstrato de dados é um conjunto de valores com seu comportamento definido por operações implementadas na forma de funções.
- É construído a partir dos tipos básicos e/ou dos tipos estruturados da linguagem.
- São entidades puramente teóricas, usadas para simplificar a descrição de algoritmos abstratos, classificar e avaliar estruturas de dados e descrever formalmente certos tipos de sistemas.



- Tanto a representação quanto as operações do TAD são especificadas pelo programador
- Para a criação de um TAD é essencial ocultar os dados do usuário, ou seja, deve-se tornar invisível a sua implementação para o usuário.
- O usuário não tem acesso direto à informação lá armazenada.
- O usuário utiliza o TAD como uma caixa-preta por meio de sua interface.
- A estratégia de ocultação de informações permite a implementação e a manutenção de módulos sem afetar os programas do usuário.

- Quando se define um TAD, se está preocupado com o que ele faz e não em como ele faz.
- Na linguagem C, é comumente implementado na forma de dois módulos
 - Implementação
 - Implementa as funcionalidades do TAD e não é visível pelo usuário;
 - Interface
 - Apenas declara as funções que correspondem às operações do TAD e é visível pelo usuário;

- Encapsulamento

- Ao se ocultar a implementação, é fornecido um conjunto de operações possíveis para o TAD.
- Isso é tudo o que o usuário precisa saber para fazer uso do TAD.
- O usuário não precisa de nenhum conhecimento técnico sobre como a implementação trabalha para usá-lo.
- Utilização simplificada.

- Segurança

- O usuário não tem acesso direto aos dados, isso evita que ele manipule os dados de maneira imprópria.

- Flexibilidade

- Pode-se alterar o TAD sem alterar as aplicações que o utilizam.
- Pode-se ter diferentes implementações de um TAD desde que todas respeitem a mesma interface.

- Reutilização

- A implementação do TAD é feita em um módulo diferente do programa do usuário.

Operações Básicas

Estruturas de
Dados

Fábio Duncan

Conceitos
Básicos

Vantagens de
Uso

**Operações
Básicas**

Exemplo - O
tipo File

Tipo Opaco

Exemplo de
Imple-
mentação

Pilhas

Filas

Referências

- Tipos abstratos de dados incluem as operações para a manipulação de seus dados.
- Essas operações variam de acordo com o TAD criado, porém algumas operações comumente implementadas são:
 - Criação do TAD
 - Inserção de um novo elemento no TAD
 - Remoção de um elemento do TAD
 - Acesso a um elemento do TAD
 - Destruição do TAD

Exemplo

O tipo file

Estruturas de Dados

Fábio Duncan

Conceitos Básicos

Vantagens de Uso

Operações Básicas

Tipo Opaco

Exemplo de Imple- mentação

Pilhas

Filas

Referências

- O tipo FILE é uma estrutura que contém as informações sobre um arquivo ou fluxo de texto necessário para realizar as operações de entrada ou saída sobre ele
- Exemplos de conteúdo do tipo file
 - Descritor do arquivo
 - Posição atual
 - Indicador de fim de arquivo
 - Indicador de erro
- O conteúdo da estrutura pode mudar dependendo da implementação.

Exemplo

O tipo file

Estruturas de Dados

Fábio Duncan

Conceitos Básicos

Vantagens de Uso

Operações Básicas

Tipo Opaco

Exemplo de Imple- mentação

Pilhas

Filas

Referências

- Para se trabalhar com arquivos em linguagem C declara-se um ponteiro para o arquivo
 - FILE* arquivo;
 - Desse modo, o usuário possui apenas um ponteiro para onde os dados estão armazenados, mas não pode acessá-los diretamente.
 - A maneira de acessar o conteúdo do ponteiro FILE é por meio das operações definidas em sua interface.
- Exemplos de operações de manipulação de arquivos providas pelo tipo file
 - fopen()
 - fclose()
 - fscanf()
 - fprintf()
 - feof()

Tipo Opaco

Estruturas de
Dados

Fábio Duncan

Conceitos
Básicos

Vantagens de
Uso

Operações
Básicas

Exemplo - O
tipo File

Tipo Opaco

Exemplo de
Imple-
mentação

Pilhas

Filas

Referências

- Um tipo de dado opaco é um tipo de dado cuja estrutura de dados concreta não está definida em uma interface.
- Possui informações ocultas, já que seus valores só podem ser manipulados ao chamar sub-rotinas que tenham acesso às informações que faltam.
- A representação concreta do tipo está oculta aos usuários e a implementação visível está incompleta.
- São frequentemente usados para implementar tipos de dados abstratos.
- Em resumo, o tipo opaco representa uma forma de esconder os detalhes de sua implementação dos programadores que apenas farão uso do módulo ou biblioteca.

- Para criar um tipo opaco na linguagem C, utiliza-se dois arquivos
 - Arquivo ".c"
 - Declaração dos tipos de dados que ficarão ocultos do usuário do TAD.
 - Implementação das funções.
 - Arquivo ".h"
 - Declara os tipos que irão representar os dados ocultos do arquivo ".c" e que somente poderá ser declarado pelo usuário na forma de um ponteiro.
 - São declarados os protótipos das funções visíveis para o usuário.



Exemplo de Implementação de um TAD

Programa Exemplo 1 - Ponto no plano cartesiano

Estruturas de
Dados

Fábio Duncan

Conceitos
Básicos

Vantagens de
Uso

Operações
Básicas

Exemplo - O
tipo File

Tipo Opaco

Exemplo de
Imple-
mentação

Pilhas

Filas

Referências

- Faça um programa que permita a criação de pontos em um plano cartesiano. Cada ponto deverá ser implementado como um tipo abstrato de dados. O tipo abstrato ponto deverá conter as seguintes funcionalidades:
 - 1 Criação de um ponto
 - 2 Destruição de um ponto
 - 3 Consulta às coordenadas de um ponto
 - 4 Cálculo da distância entre dois pontos

Exemplo de Implementação de um TAD

Programa Exemplo 1 - Ponto no plano cartesiano

Estruturas de Dados

Fábio Duncan

Conceitos Básicos

Vantagens de Uso

Operações Básicas

Exemplo - O tipo File

Tipo Opaco

Pilhas

Filas

Referências

- Arquivo de Cabeçalho

```
1 typedef struct reg_ponto tipo_ponto;
2
3 tipo_ponto *criar_ponto(float, float);
4 void consultar_coordenada(tipo_ponto*, float*, float*);
5 float calcular_distancia(tipo_ponto*, tipo_ponto*);
```




Exemplo de Implementação de um TAD

Programa Exemplo 1 - Ponto no plano cartesiano

● Exemplo de uso do TAD ponto

```
1  #include <stdio.h>
2  #include "ponto.h"
3
4  int main()
5  {
6      tipo_ponto *ponto1, *ponto2, *ponto3;
7      float x,y;
8      printf("Entre com o valor para x:");
9      scanf("%f", &x);
10     printf("Entre com o valor para y:");
11     scanf("%f", &y);
12     ponto1 = criar_ponto(x, y);
13     ponto2 = criar_ponto(20, 20);
14     ponto3 = criar_ponto(2, 3);
15
16     consultar_coordenada(ponto2, &x, &y);
17     printf("\nPonto2 (%.2f,%.2f)", x, y);
18     consultar_coordenada(ponto1, &x, &y);
19     printf("\nPonto1 (%.2f,%.2f)", x, y);
20     printf("Distancia entre o ponto1 e o ponto2: %f", calcular_distancia(ponto1, ponto2));
21     return 0;
22 }
23
```



Exemplo de Implementação de um TAD

Programa Exemplo 1 - Ponto no plano cartesiano

● Implementação do TAD ponto - Parte 1

```
1 #include <stdlib.h>
2 #include <math.h>
3
4 struct reg_ponto
5 {
6     float x;
7     float y;
8 };
9
10 typedef struct reg_ponto tipo_ponto;
11
12 tipo_ponto *criar_ponto(float x, float y)
13 {
14     tipo_ponto *ponto;
15     ponto = (tipo_ponto*) malloc(sizeof(tipo_ponto));
16     ponto->x = x;
17     ponto->y = y;
18     return ponto;
19 }
20
```



Exemplo de Implementação de um TAD

Programa Exemplo 1 - Ponto no plano cartesiano

Estruturas de
Dados

Fábio Duncan

Conceitos
Básicos

Vantagens de
Uso

Operações
Básicas

Exemplo - O
tipo File

Tipo Opaco

Exemplo de
Imple-
mentação

Pilhas

Filas

Referências

● Implementação do TAD ponto - Parte 2

```
1 void consultar_coordenada(tipo_ponto *ponto, float *x, float *y)
2 {
3     *x = ponto->x;
4     *y = ponto->y;
5 }
6
7 float calcular_distancia(tipo_ponto *ponto1, tipo_ponto *ponto2)
8 {
9     float distancia_x, distancia_y;
10
11     distancia_x = ponto2->x - ponto1->x;
12     distancia_y = ponto2->y - ponto1->y;
13     return sqrt(pow(distancia_x, 2) + pow(distancia_y, 2));
14 }
15
```

Tipo Abstrato de Dados Pilha - Stack

Conceitos Básicos

Estruturas de
Dados

Fábio Duncan

Conceitos
Básicos

Vantagens de
Uso

Operações
Básicas

Exemplo - O
tipo File

Tipo Opaco

Exemplo de
Implementação

Pilhas

Filas

Referências

- É uma lista linear em que todas as inserções, retiradas e, geralmente, todos os acessos são feitos em apenas um extremo da lista.
- Uma pilha (stack) é uma versão de uma lista encadeada com restrições.
- Os novos nós só podem ser adicionados no topo da pilha e também só podem ser removidos os nós do topo de uma pilha.
- Intuitivamente pode ser representado por um monte de pratos em uma prateleira, sendo conveniente retirar ou adicionar pratos na parte superior.
- A pilha é conhecida como uma estrutura de dados do tipo último a entrar, primeiro a sair (last-in,first-out, ou LIFO)

Tipo Abstrato de Dados Pilha - Stack

Conceitos Básicos

Estruturas de
Dados

Fábio Duncan

Conceitos
Básicos

Vantagens de
Uso

Operações
Básicas

Exemplo - O
tipo File

Tipo Opaco

Exemplo de
Imple-
mentação

Pilhas

Filas

Referências

- A referência a uma pilha é feita por meio de um ponteiro para o elemento do topo da pilha.
- A ligação existente no último nó da pilha é definida com NULL para indicar o final da pilha.
- As pilhas e as listas encadeadas são representadas de maneira idêntica.
- A diferença entre pilhas e listas encadeadas é que as inserções e remoções podem ocorrer em qualquer lugar de uma lista encadeada, mas apenas no topo de uma pilha.
- As duas principais funções utilizadas para manipular pilhas são:
 - Empilhar (push): Cria um novo nó e o coloca no topo da pilha.
 - Desempilhar (pop): Remove o nó do topo de uma pilha, libera a memória que estava alocada ao nó removido e retorna o valor removido.



Tipo Abstrato de Dados Pilha - Stack

Conceitos Básicos

Estruturas de
Dados

Fábio Duncan

Conceitos
Básicos

Vantagens de
Uso

Operações
Básicas

Exemplo - O
tipo File

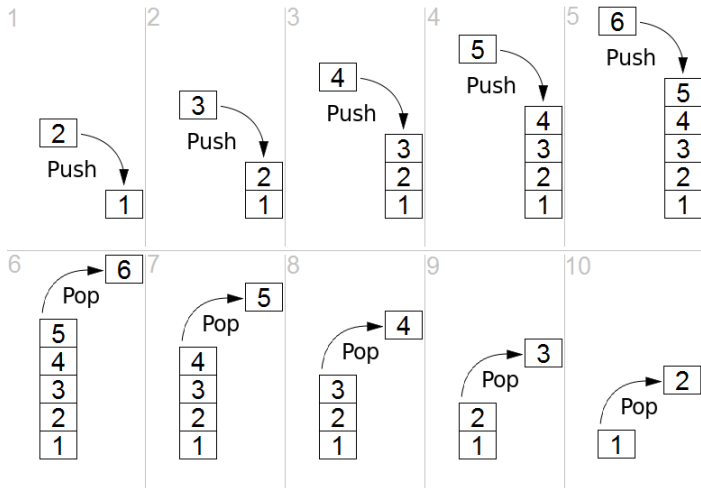
Tipo Opaco

Exemplo de
Imple-
mentação

Pilhas

Filas

Referências





Tipo Abstrato de Dados Pilha - Stack

Aplicações

Estruturas de
Dados

Fábio Duncan

Conceitos
Básicos

Vantagens de
Uso

Operações
Básicas

Exemplo - O
tipo File

Tipo Opaco

Exemplo de
Imple-
mentação

Pilhas

Filas

Referências

- Funções recursivas em compiladores.
- Mecanismo de desfazer/refazer dos editores de texto.
- Situações em que é preciso “guardar para mais tarde” vários elementos e “lembrar” sempre do último elemento armazenado.



Tipo Abstrato de Dados Fila - Queue

Conceitos Básicos

Estruturas de
Dados

Fábio Duncan

Conceitos
Básicos

Vantagens de
Uso

Operações
Básicas

Exemplo - O
tipo File

Tipo Opaco

Exemplo de
Imple-
mentação

Pilhas

Filas

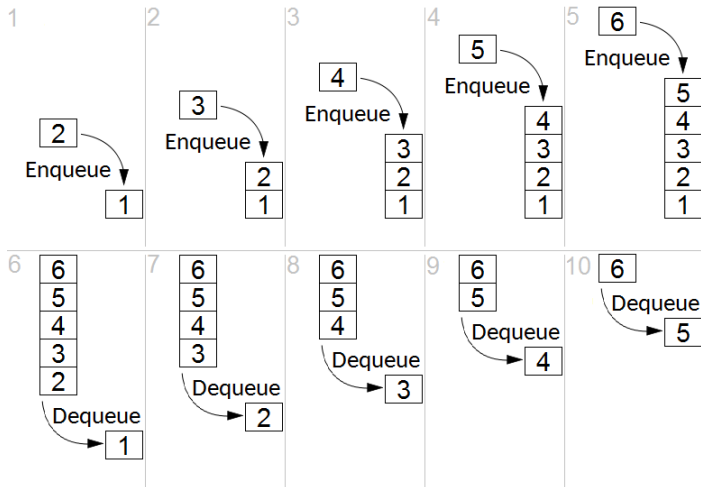
Referências

- É uma lista linear em que os nós são removidos apenas do início da estrutura e são inseridos apenas em seu final.
- É uma estrutura de dados do tipo primeiro a entrar, primeiro a sair (first-in, first-out ou FIFO).
- As duas principais funções utilizadas para manipular pilhas são:
 - Enfileirar (enqueue): Cria um novo nó e o coloca no final da fila.
 - Desenfileirar (dequeue): Remove o nó do início da fila, libera a memória que estava alocada ao nó removido e retorna o valor removido.



Tipo Abstrato de Dados Fila - Queue

Conceitos Básicos



Tipo Abstrato de Dados Fila - Queue

Aplicações

Estruturas de Dados

Fábio Duncan

Conceitos Básicos

Vantagens de Uso

Operações Básicas

Exemplo - O tipo File

Tipo Opaco

Exemplo de Imple- mentação

Pilhas

Referências

- Implementação de Filas de Processos em Sistemas Operacionais.
- Suportar armazenamento de dados (spooling) para impressão.
- Os pacotes de informações esperam em filas em redes de computadores.



André Ricardo Backes.

Estrutura de Dados Descomplicada : em Linguagem C.
Elsevier, 2016.



Harvey M Deitel and Paul J Deitel.

Como programar em C.
LTC, 1999.