

Estrutura de Dados

Diógenes Carvalho Matias

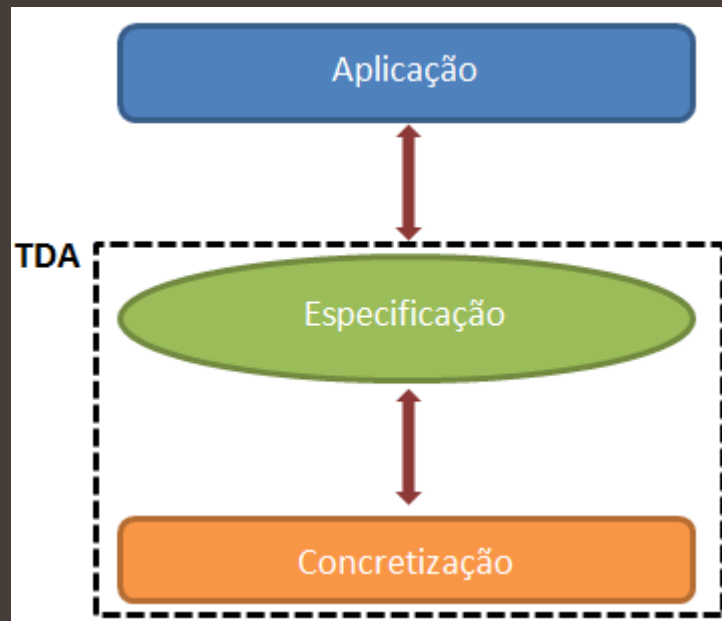
Introdução

Tipo de Dado Abstrato

O conceito de abstração de dados, proposto por Barbara Liskov em 1974, tem a mesma ideia da abstração criada por funções. Diante disso, o objetivo é definir um tipo de dado que possa ser independente de qualquer implementação específica e separado do utilizador da aplicação. Dito de outra maneira, um tipo de dado abstrato é caracterizado por definir um conjunto de valores e de operações para manipular esses valores.

Introdução

Tipo de Dado Abstrato



Introdução

Módulos

Um módulo na linguagem C pode ser criado utilizando arquivos de cabeçalho e de implementação. Cabe ressaltar que um módulo não é um tipo abstrato de dados, pois pode ser utilizado para outros fins:

Introdução

Vamos ver a Linguagem C

Introdução

Linguagem C

A primeira versão de C foi criada por Dennis Ritchie em 1972 nos laboratórios Bell para ser incluído como um dos softwares a serem distribuídos juntamente com o sistema operacional Unix do computador PDP-11, na equipe certificada por Ken Thompson.

Introdução

Linguagem C

Ao ponto de vista técnico, o surgimento do C iniciou com a linguagem ALGOL 60, definida em 1960. ALGOL era uma linguagem de alto nível, que permitia ao programador trabalhar "longe da máquina", sem se preocupar com os aspectos de como cada comando ou dado era armazenado ou processado. Foi criado para substituir o FORTRAN. ALGOL não teve sucesso, talvez por tentar ser de muito alto nível em uma época em que a maioria dos sistemas operacionais exigiam do usuário um grande conhecimento de hardware.

Introdução

Linguagem C

O mais interessante desta versão de C era que os programas-fonte criados para rodar em um tipo de computador podiam ser transportados e recompilados em outros sem grandes problemas. A esta característica dá-se o nome de portabilidade. Com ela, uma empresa que desenvolve um programa pode fazê-lo rodar em diferentes computadores sem ter um elevado custo a cada vez que isto for feito.

Em 1985, ANSI (American National Standards Institute) estabeleceu um padrão oficial de C o chamado "C ANSI" .

Introdução

Módulos

Arquivo de cabeçalho: Os arquivos '.h' são utilizados para especificar assinatura de funções, definições de constantes, tipos de dados criados pelo usuário etc. De modo geral, os arquivos de cabeçalho tem como função definir a interface de um módulo.

Introdução

Módulos

Arquivo de implementação: Os arquivos ‘.c’ implementam as funções definidas na interface. De modo geral, esses arquivos são compostos por diversas funções e estruturas de dados utilizadas internamente.

Introdução

Módulos

Arquivo de implementação: Os arquivos ‘.c’ implementam as funções definidas na interface. De modo geral, esses arquivos são compostos por diversas funções e estruturas de dados utilizadas internamente.

Estrutura de Dados

Sintaxe da linguagem

Estrutura básica da linguem:

```
#include <stdio.h>
```

```
int main() {
```

```
    /* Meu primeiro programa em C */
```

```
    printf("Boa noite , Diogenes Carvalho Matias! \n");
```

```
    return 0;
```

```
}
```



UNIFG
LAUREATE INTERNATIONAL UNIVERSITIES

Estrutura de Dados

Sintaxe da linguagem

Estrutura de bibliotecas básica da linguagem:

```
#include <stdio.h> //standard input-output (biblioteca  
padrão de entrada/saída).
```

```
#include <stdlib.h> //standard library (biblioteca de  
propósito geral)
```

```
//Corpo de execução do programa em C  
int main (int argc, char *argv[ ]) {
```

```
//corpo principal do programa
```

```
}
```



Estrutura de Dados

Sintaxe da linguagem

Declaração de variáveis da linguagem:

`int a; //declaração de variáveis globais`

Tipo de primitivos:

Tipos Primitivos		
Tipo	Tamanho	Valores válidos
char	1 byte	-128 a 127
unsigned char	1 byte	0 a 255
short int	2 bytes	-32.768 a 32.767
unsigned short int	2 bytes	0 a 65.535
long int	4 bytes	-2.147.483.648 a 2.147.483.647
unsigned long int	4 bytes	0 a 4.294.967.295
float	4 bytes	10^{-38} a 10^{38}
double	8 bytes	10^{-308} a 10^{308}

Estrutura de Dados

Sintaxe da linguagem

Declarações de funções e procedimentos da linguagem:

Funções:

```
float funcao_de_exemplo_basica() {  
    return 10;  
}
```

Procedimento:

```
void procedimento_exemplo (float f) {  
    f = 10;  
}
```



Estrutura de Dados

Sintaxe da linguagem

Qual a diferença entre função e procedimento?

Estrutura de Dados

Operadores da linguagem:

Operadores			
Tipo	Operador	Propósito	Exemplo
Aritméticos	+	Adição	<code>a = 4 + 1; // 5</code>
	-	Subtração	<code>a = 4 - 1; // 3</code>
	*	Multiplicação	<code>a = 2 * 4; // 8</code>
	/	Divisão	<code>a = 8 / 2; // 4</code>
	%	Módulo (resto da divisão)	<code>a = 5 % 2; // 1</code>
Atribuição	=	Atribuição simples	<code>a = 50;</code>
Lógicos	&&	"e" lógico	<code>(a > 1) && (b < 1)</code>
		"ou" lógico	<code>(a > 1) (b < 1)</code>
	!	não (inversão)	<code>!(a > 2)</code>
Relacionais (Comparação)	==	igual a	<code>(a == 0)</code>
	!=	diferente de	<code>(a != 0)</code>
	<	menor que	<code>(a < 0)</code>
	>	maior que	<code>(a > 0)</code>
	<=	menor ou igual a	<code>(a <= 0)</code>
	>=	maior ou igual a	<code>(a >= 0)</code>
Incremento e Decremento	++	Incremento	<code>a++;</code>
	--	Decremento	<code>a--;</code>



Estrutura de Dados

Funções de entrada:

```
printf(formato, argumentos);
```

Função para saída de valores segundo um determinado formato.

Exemplo:

```
printf(“%d %g\n”, 65, 7.3);
```

```
printf(“Inteiro = %d Real = %f\n”, 65, 7.3);
```

Estrutura de Dados

Funções de saída:

`scanf(formato, lista de endereços).`

Função para capturar e armazenar valores fornecidos via teclado.

Exemplo:

```
scanf("%d",&n);
```

```
scanf("%d",&numero);
```

```
scanf("%d:%d",&h,&m);
```

Estrutura de Dados

Especificadores de formatos:

%c	char
%d	int
%u	unsigned int
%f	double ou float
%e	double ou float (científico)
%s	cadeia de caracteres
\n	quebra de linha
\t	tabulação
\"	caractere "
\\	caractere \

Estrutura de Dados

Comandos da linguagem:

Comando	Propósito	Sintaxe
Declaração de variável	Declaração de variável	<code>tipo nome_variavel = valor_inicial;</code>
Declaração de constante	Declaração de constante	<code>#define NOME_CONSTANTE valor</code>
Bloco	Marcar um bloco de cód.	<code>{ } //Abre e fecha chaves "{}"</code>

Estrutura de Dados

Comandos da linguagem:

if	Comando condicional	<pre>if (a > b) { printf("%s", "A é maior que B"); } else { printf("%s", "A é igual ou menor que B"); }</pre>
-----------	---------------------	--

Exemplo pratico:

```
#include <stdio.h>  
//Ordena as variáveis a e b  
main( ) {  
    int a=7, b = 2, x;  
    if (a > b){  
        x = a;  
        a = b;  
        b = x;  
    }  
    if (a != b) {  
        printf("diferente\n");  
    }  
}
```



Estrutura de Dados

Comandos da linguagem:

switch	Comando condicional	<pre>switch (i) { case 0 : printf("%s", "ZERO"); break; case 1: printf("%s", "UM"); break; case 2: printf("%s", "DOIS"); break; }</pre>
---------------	---------------------	---

Estrutura de Dados

Comandos da linguagem:

Exemplo:

```
#include<stdio.h>
```

```
main(){
```

```
    char opcao;
```

```
    printf("Digite uma opcao( y ou z ):");
```

```
    scanf("%c", &opcao);
```

```
        switch (opcao) {
```

```
            case 'y':
```

```
                printf("inserir\n");
```

```
            break;
```

```
            case 'z':
```

```
                printf("remover\n");
```

```
            break;
```

```
            default:
```

```
                printf("opcao invalida!\n");
```

```
        }
```

```
    }
```



UNIFG
LAUREATE INTERNATIONAL UNIVERSITIES

Estrutura de Dados

Exercícios para ser feliz XD:

Estrutura de Dados

Exercicios para ser feliz XD:

Exercicio level 01:

1-Escreva um algoritmo para ler o salário mensal atual de um funcionário e o percentual de reajuste. Calcular e escrever o valor do novo salário.

2-O custo de uma moto nova ao consumidor é a soma do custo de fábrica com a porcentagem do distribuidor e dos impostos (aplicados ao custo de fábrica). Supondo que o percentual do distribuidor seja de 28% e os impostos de 45%, escrever um algoritmo para ler o custo de fábrica de uma moto, calcular e escrever o custo final ao consumidor.

3-Escreva um algoritmo que armazene o valor 20 em uma variável X e o valor 40 em uma variável Y. A seguir (utilizando apenas atribuições entre variáveis) troque os seus conteúdos fazendo com que o valor que está em X passe para Y e vice-versa. Ao final, escrever os valores que ficaram armazenados nas variáveis.

Estrutura de Dados

Exercícios para ser feliz XD:

Exercício level 02:

4- Escreva um algoritmo para ler um valor (do teclado) e escrever (na tela) o seu antecessor.

5-Escreva um algoritmo para ler as dimensões de um retângulo (base e altura), calcular e escrever a área do retângulo.

6- Faça um algoritmo que leia a idade de uma pessoa expressa em anos, meses e dias e escreva a idade dessa pessoa expressa apenas em dias. Considerar ano com 365 dias e mês com 30 dias.

7- Escreva um algoritmo para ler o salário mensal atual de um funcionário e o percentual de reajuste. Calcular e escrever o valor do novo salário.