

UNIVERSIDADE FEDERAL DE VIÇOSA
CAMPUS FLORESTAL

TRABALHO PRÁTICO - AEDS1

DENER VIEIRA RIBEIRO - 3872
GERMANO BARCELOS DOS SANTOS - 3873

Prof. Thais R. M. Braga Silva
Algoritmos e Estruturas de Dados I

FLORESTAL - MG
2019

Foram definidos padrões de desenvolvimento para ficar mais fácil e simples a colaboração entre os desenvolvedores:

- VARIÁVEIS:
 - Nome: sempre iniciadas com letra minúscula e usar nomes significativos.
Ex.: nomeDaVariavel
- STRUCTS:
 - Nome: nome iniciado com letra minúscula e final do nome com um “_t”.
Ex.: nomeDaStruct_t
- TIPOS DE DADOS:
 - Nome: novos tipos de dados criados por typedef iniciados com letra maiúscula. Ex.: NovoTipoDeDados
- FUNÇÕES:
 - Nome: funções relacionadas a tipos devem iniciar com o nome do tipo e em seguida o sufixo de sua função no sistema (os nomes se iniciam com letra minúscula). Ex.: A função de somar um tipo MeuTipo deve ter o formato meuTipoSoma()
 - Criar:
 - Usar o sufixo Init. Ex.: meuTipoInit()
 - Ler:
 - Usar o sufixo ToString. Ex.: meuTipoToString()

- Ou usar o sufixo Get (para retornar uma variavel). Ex.: meuTipoGet()
- Ou usar o sufixo Get + <Dado>. Ex.: meuTipoGetValor(), meuTipoGetEndereco().

- Inserir:
 - Usar o sufixo Insert. Ex.: meuTipoInsert()

- Remover:
 - Usar o sufixo Remove. Ex.: meuTipoRemove()

- Destruir:
 - Usar o sufixo Free. Ex.: meuTipoFree()

- BIBLIOTECAS:
 - File.h:
 - Inclusão de cabeçalhos NECESSÁRIOS para a biblioteca e protótipos
 - File.c:
 - Incluir APENAS “File.h” e o desenvolvimento das funções.

Após serem definidos os padrões de desenvolvimento, foi pensado nos TADs(Tipos Abstratos de Dados) que seriam necessários para fazer o sistema. Na tabela abaixo encontram-se os TADs utilizados junto com sua breve descrição, atributos e operações.

TAD	DESCRIÇÃO	ATRIBUTOS	OPERAÇÕES
Date	Armazena os	Dia, mes e ano	Criar, transformar no

	atributos de data		n-ésimo dia do ano, criar a partir do n-ésimo dia do ano, validar e transformar em string
Shopping	Armazena os atributos de uma compra.	Data e quantidade	Criar, validar e transformar em string
ShoppingList	Estrutura responsável por fazer uma lista de compras usando lista encadeada.	Compra, apontador para o próximo bloco da estrutura.	Criar, inserir, concatenar, contar produtos, contar blocos da estrutura, transformar e string e desalocar.
Node	Representa um item da matriz esparsa.	Lista de compra, valores de linha e coluna (posição) na matriz esparsa, apontadores para o próximo item a direita e abaixo.	Criar, desalocar, concatenar, comparar e validar.
SpMatrix	Estrutura responsável por fazer as operações sobre os Nodes (itens da matriz esparsa.)	Item cabeça da matriz e valores máximos de linha e coluna.	Criar, inserir, transformar em string, desalocar, gerar os valores de compras por produto e compras por cliente

			em forma de string, e validar.
--	--	--	--------------------------------

Além disso, foi feito um módulo para fazer a leitura e criar uma matriz esparsa a partir de um arquivo informado pelo usuário.