

Estruturas de Dados

Prof. Rodrigo Martins

rodrigo.martins@francomontoro.com.br

Cronograma da aula inaugural

- Apresentação da disciplina
 - Objetivos
 - Ementa
 - Critérios de avaliação
 - Metodologia de Trabalho
 - Bibliografias
- Introdução as estruturas de dados
- Git x GitHub
- Exercícios de revisão

Objetivos da disciplina

- Apresentar ao aluno o conceito de abstração de dados, sua importância para os princípios de modularidade, encapsulamento e independência de implementação.
- Apresentar as estruturas de dados clássicas, suas características funcionais, formas de representação, operações associadas e complexidade das operações.
- Ao final da disciplina, o aluno estará capacitado a selecionar as estruturas de dados, os algoritmos de classificação de dados e as respectivas representações que sejam mais adequadas a uma dada aplicação, implementando-as com uso dos recursos de linguagem de programação mais apropriados ao caso.

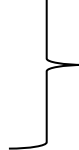
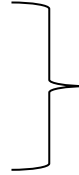
Ementa

- Armazenamento e recuperação aleatória: vetores, matrizes, registros, alocação dinâmica, ordenação para otimização de busca, algoritmos (busca, inserção, remoção e alteração).
- Armazenamento e recuperação na mesma ordem: fila, implementação, algoritmos.
- Armazenamento e recuperação em ordem inversa: pilha, implementação, algoritmos.
- Noções de memória secundária: acesso sequencial e acesso aleatório.

Metodologias de Trabalho

- Material exposto em sala de aula (Apresentações);
- Indicação de Sites sobre o conteúdo (Artigos);
- Exemplos e Exercícios práticos;
- Uso de metodologias ativas para desenvolvimento de projetos.

Critérios de Avaliação

- T1 – Lista de Exercícios (30%)
 - P1 - Avaliação Bimestral (70%)
- 
- MB1
-
- T2 - Lista de Exercícios (30%)
 - P2 – Avaliação Bimestral (70%)
- 
- MB2

Média Final – $(MB1 + MB2) / 2$

Critérios de Avaliação

- Média final maior ou igual a 7,0 (sete) implicará em aprovação sem exame final;
- Média final igual ou superior a 4,0 (quatro) e inferior a 7,0 (sete) dependerá de aprovação em exame final;
- Média final de aproveitamento inferior a 4,0 (quatro) implicará em reprovação;
- A aprovação em exame final será obtida se a média aritmética da média final de aproveitamento com a nota do exame final for igual ou superior a 5,0 (cinco).

Critérios de Avaliação

- Além do requisito nota, para aprovação é necessário que o discente tenha frequência maior ou igual a 75%.
- Caso a frequência seja inferior a 75%, o discente será reprovado por falta, independente de estar aprovado por nota, e portanto, sem a possibilidade de realização de exame.

Bibliografia Básica

- VELOSO, P. et Alli. Estruturas de Dados. Ed. Campus, 1986.
- TANENBAUM, A., LANGSAM, Y., AUGUSTEIN, M. Estruturas de Dados Usando C. Makron Books, 1995.
- PEREIRA, Silvio do Lago. Estrutura de Dados Fundamentais: conceitos e aplicações. Editora Erica. São Paulo. 1996.

Bibliografia Complementar

- VILLAS, Marcos Vianna. Estrutura de Dados: conceitos e técnicas de implementação. Editora Campus. Rio de Janeiro. 1993.
- SAVARCFILTER, Jayme Luiz. Estrutura de Dados e seus Algoritmos. LTC. Rio de Janeiro. 1994.
- WIRTH, NIKLAUS. Algoritmos e Estruturas de Dados. Prentice-Hall do Brasil, Rio de Janeiro, 1999.

O que são estruturas de dados e por que são importantes

- Estruturas de dados são formas organizadas de armazenar, gerenciar e acessar dados de forma eficiente.
- Elas determinam como os dados são dispostos na memória e como interagimos com eles.

O que são estruturas de dados e por que são importantes

- **Importância 1**
 - Eficiência:
 - Permitem otimizar o uso de memória e tempo de processamento. Exemplo: acessar elementos em um array é mais rápido que em uma lista encadeada.

O que são estruturas de dados e por que são importantes

- **Importância 2**
 - Organização:
 - Facilitam o gerenciamento de grandes volumes de dados, como em bancos de dados ou sistemas complexos.

O que são estruturas de dados e por que são importantes

- **Importância 3**
 - Base para algoritmos:
 - Muitos algoritmos são criados para operar sobre estruturas de dados específicas. Sem elas, resolver problemas computacionais seria ineficaz.

Relação entre estruturas de dados e algoritmos

- **Interdependência**
 - Estruturas de dados e algoritmos andam de mãos dadas.
 - A escolha da estrutura de dados adequada influencia diretamente a eficiência de um algoritmo.
 - Exemplo:
 - Para buscas rápidas, uma tabela hash é mais adequada do que uma lista.

Relação entre estruturas de dados e algoritmos

- **Soluções otimizadas**
 - Um algoritmo eficiente pode não ser útil se a estrutura de dados escolhida for inadequada.
 - Por outro lado, uma boa estrutura de dados pode simplificar a implementação do algoritmo.

Aplicações práticas e exemplos do mundo real

- **Redes sociais**
 - Estruturas de grafos representam conexões entre pessoas (amigos no Facebook, seguidores no Twitter).
- **Motores de busca**
 - Árvore B-trees são usadas em índices de bancos de dados e sistemas de arquivos para buscas rápidas.

Aplicações práticas e exemplos do mundo real

- **Jogos**
 - Pilhas e filas são utilizadas para implementar movimentos do jogador, desfazer ações ou controlar a IA dos oponentes.
- **Mapas e GPS**
 - Grafos representam ruas e rotas. Algoritmos como Dijkstra utilizam essas estruturas para encontrar o caminho mais curto.

Aplicações práticas e exemplos do mundo real

- **E-commerce**
 - Tabelas hash ajudam a mapear itens no carrinho de compras ou fazer busca de produtos com alta eficiência.

Conceitos Básicos: dados, tipos de dados e organização

- **Dados**
 - São informações que podem ser armazenadas, manipuladas e processadas por um sistema computacional.
 - Exemplos:
 - números, caracteres, textos, imagens, áudio, vídeos, entre outros.

Conceitos Básicos: dados, tipos de dados e organização

- **Tipos de dados**
 - Primitivos
 - Fornecidos pela linguagem de programação, como inteiros, floats, caracteres e booleanos.
 - Compostos
 - Criados a partir dos tipos primitivos, como arrays, structs (em C) e objetos (em linguagens orientadas a objetos).
 - Abstratos
 - Conjuntos de operações definidas para manipular dados. Exemplos: pilhas, filas, listas, árvores e grafos.

Conceitos Básicos: dados, tipos de dados e organização

- **Organização**
 - A organização dos dados no sistema é fundamental para acessar e processá-los de forma eficiente.

Conceitos Básicos: dados, tipos de dados e organização

- **Formas comuns de organização**
 - Sequencial (linear)
 - Os dados são organizados em sequência, como em arrays e listas.
 - Hierárquica
 - Os dados são organizados em uma estrutura em forma de árvore, como árvores binárias.
 - Relacional
 - Dados organizados em relações, como tabelas em um banco de dados.
 - Associativa
 - Usa pares de chave-valor para rápida recuperação, como em tabelas hash.

Ferramenta de Apoio

- CodeBlocks
- <https://sourceforge.net/projects/codeblocks/files/Binaries/20.03/Windows/codeblocks-20.03mingw-setup.exe>

Git vs. GitHub

- Artigo Alura
 - <https://encurtador.com.br/bjHWc>

Exercícios Propostos

1. Elabore um programa em C++ que solicite o peso e a altura de uma determinada pessoa. Após a digitação, exibir se esta pessoa está ou não com seu peso ideal, conforme tabela abaixo:

IMC ($\text{IMC} = \text{peso} / \text{altura}^2$)	MENSAGEM
$\text{imc} < 20$	Abaixo do peso
$20 \geq \text{imc} < 25$	Peso Ideal
$\text{IMC} \geq 25$	Acima do Peso

2. Elaborar um programa em C++ em que dada a idade de um nadador, classifique-o em uma das seguintes categorias: infantil A (de 5 a 7 anos), infantil B (de 8 a 10 anos), juvenil A (de 11 a 13 anos), juvenil B (14 a 17 anos) e senior (maior que 17 anos)

Exercícios Propostos

3. Faça um programa em C++ que receba o número de horas trabalhadas e o valor do salário mínimo. Calcule e mostre o salário a receber seguindo as regras abaixo:
 - a. A hora trabalhada vale a metade do salário mínimo;
 - b. O salário bruto equivale ao número de horas trabalhadas multiplicado pelo valor da hora trabalhada
 - c. O imposto equivale a 3% do salário bruto;
 - d. O salário a receber equivale ao salário bruto menos o imposto.
4. Construa um programa em C++ que calcule o novo salário de um funcionário. Considere que o funcionário deverá receber um reajuste de 15% caso seu salário seja menor que 1000. Se o salário for maior ou igual a 1000, mas menor ou igual a 1500, o reajuste deve ser de 10%. Caso o salário seja maior que 1500, o reajuste deve ser de 5%.

Exercícios Propostos

5. Construa um programa em C++ que calcule e apresente quanto deve ser pago por um produto considerando a leitura do preço de etiqueta (PE) e o código da condição de pagamento (CP). Utilize para os cálculos a tabela de condições de pagamento a seguir:

Código da condição de pagamento	Condição de pagamento
1	À vista em dinheiro ou cheque, com 10% de desconto
2	À vista com cartão de crédito, com 5% de desconto
3	Em 2 vezes, preço normal de etiqueta sem juros
4	Em 3 vezes, preço de etiqueta com acréscimo de 10%

Exercícios Propostos

6. Escreva um programa em C++ que mostre o quadrado dos números inteiros no intervalo de 1 a 20.
7. Escreva um programa em C++ que escreva todos os números múltiplos de 5, no intervalo de 1 a 500.
8. Em uma eleição presidencial existem dois candidatos. Os votos são informados através de códigos. Os dados utilizados para a contagem dos votos têm-se a seguinte codificação: 1,2= voto para os respectivos candidatos; 3= voto nulo; 4= voto em branco; Elabore um programa em C++ que leia o código do candidato em um voto. Como finalizador do conjunto de votos, tem-se o valor 0. Calcule e escreva: (1) percentual de votos para cada candidato; (2) percentual de votos nulos; (3) percentual de votos em branco;
9. Faça um programa em C++ que leia dez conjuntos de dois valores, o primeiro representando o número do aluno e o segundo a sua altura em centímetros. Encontre o aluno mais alto e o mais baixo. Exiba o número do aluno mais baixo o número de aluno mais alto e as respectivas alturas.

Exercícios Propostos

10. Em um cinema, certo dia, cada espectador respondeu a um questionário, que perguntava a sua idade (ID) e a opinião em relação ao filme (OP), seguindo os seguintes critérios:

Opinião (OP)	Significado
1	Ótimo
2	Bom
3	Regular
4	Ruim

Ao final da pesquisa será indicado quando a idade do usuário for informada como negativa (idade inexistente). Construa um programa em C++ que, lendo esses dados, calcule e apresente:

- A. Quantidade de pessoas que respondeu a pesquisa
- B. Média de idade das pessoas que responderam a pesquisa
- C. Porcentagem de cada uma das respostas

Referência desta aula

- Notas de Aula do Prof. Prof. Armando Luiz N. Delgado baseado em revisão sobre material de Prof.a Carmem Hara e Prof. Wagner Zola.
- <http://www.cplusplus.com/reference/>

Obrigado

Rodrigo