
Introdução às estruturas de dados em Java e Python

Rafael Alves da Costa

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
FATEC Carapicuíba

Aula 1 - Estrutura de dados

08/2025

Sumário

- 1** Programa do curso: Estrutura de dados
- 2** Referências
- 3** Estrutura de dados
- 4** História das Linguagens de Programação
- 5** Linguagem Compilada vs. Interpretada
- 6** Que Linguagem é o Java?
- 7** VS Code Online Configuração

Programa do curso: Estrutura de dados

Programa

- 1 Introdução às estruturas de dados em Java e Python (11/08 (Seg.))
- 2 Ferramentas de Análise de Desempenho de Algoritmos (18/08 (Seg.))
- 3 Orientação a Objetos (25/08 (Seg.))

AGOSTO	D	S	T	Q	Q	S	S
						1	2
	3	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28	29	30
	31						

1^a

2^a

3^a

4^a

Programa

- 4 Arrays, Listas simplesmente e duplamente encadeadas (1/09 (Seg.))
- 5 Pilhas, Filas e Deques (8/09 (Seg.))
- 6 Recursão e Busca Binária (15/09 (Seg.))
- 7 TAD Lista Posicional (22/09 (Seg.))
- 8 Tabela Espalhamento (Hash) (29/09 (Seg.))

SETEMBRO	D	S	T	Q	Q	S	S
	1	2	3	4	5	6	
	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
	28	29	30				

5*

6*

7*

8*

9*

Programa

- 9 Avaliação P1 (06/10 (Seg.))
- 10 Vistas da Prova P1 (13/10 (Seg.))
- 11 Árvores e Árvores binárias (20/10 (Seg.))

OUTUBRO	D	S	T	Q	Q	S	S
	1			1	2	3	4
	5	6	7	8	9	10	11
	12	13	14	15	16	17	18
	19	20	21	22	23	24	25
	26	27	28	29	30	31	

9^a

10^a

11^a

12^a

13^a

Programa

- 12 Fila de Prioridade (03/11 (Seg.))
- 13 Estrutura de Dados Heaps (10/11 (Seg.))
- 14 Árvores de Busca Binária (17/11 (Seg.))
- 15 Ordenação e Análise: Algoritmo Merge Sort e Quick Sort (26/11 (Seg.))
- 16 Ordenação e Análise: Selection Sort e Bubble Sort (?/? (Seg.)) **a definir**
- 17 Algoritmos de Grafos (?/? (Seg.)) **a definir**

NOVEMBRO	D	S	T	Q	Q	S	S
							1
2	3	4	5	6	7	8	
9	10	11	12	13	14	15	
16	17	18	19	20	21	22	
23	24	25	26	27	28	29	
30							

13^a
14^a
15^a
16^a
17^a

Programa

- 18 Avaliação P2 (01/12 (Seg.))
- 19 Vista de Provas P2 (08/12 (Seg.))
- 20 Encerramento da disciplina (12/12 (Seg.))

DEZEMBRO	D	S	T	Q	Q	S	S
		1	2	3	4	5	6
	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
	28	29	30	31	01 jan	02 jan	

18^a

19^a

20^a

Programa

■ AVALIAÇÕES:

P1 - Prova individual múltipla escolha com quatro alternativas. Vale 30%

P2 - Prova individual múltipla escolha com quatro alternativas. Vale 30%

A1 - Trabalhos individuais em aula (Conteúdo 1 - 8). Vale 20%

A2 - Trabalhos individuais em aula (Conteúdo 11 - em diante). Vale 20%

CRITÉRIO DE AVALIAÇÃO

- **Frequência Mínima:** 75%
- **Média Final:** Aprovação Direta ≥ 6.0

Informações do Curso

- Curso gratuito e introdutório de Python!
- Ideal para iniciantes e essencial para reforçar sua base.
- Quem concluir o curso recebe **0,5 ponto bônus na média final!**
- Acesse: netacad.com/courses/python-essentials-1

Referências

Referências

■ Referências bibliográficas:

- 1 DEITEL, H. M.; DEITEL, P. J. Java: como programar. 10. ed. São Paulo: Pearson, 2016
 - 2 GOODRICH, M. T.; TAMASSIA, R. Estruturas de dados e algoritmos em Java. 5. ed. Porto Alegre: Bookman, 2013
 - 3 GOODRICH, Michael T.; TAMASSIA, Roberto; GOLDWASSER, Michael H. Data structures and algorithms in Python. Hoboken: Wiley, 2013.
- Documentação JAVA
 - Documentação da versão 3 de python
 - Livro on-line - Python Fluente
 - Google colab (online)
 - Jupyter Notebook (offline)
 - VS Code (online)
 - VS Code (offline)

Estrutura de dados

Estrutura de dados

O que é uma estrutura de dados?

Uma estrutura de dados é uma forma organizada de armazenar e gerenciar dados em um sistema computacional, de modo que eles possam ser acessados, manipulados e processados de maneira eficiente. Ela define como os dados são organizados, relacionados e operados. Exemplos comuns incluem:

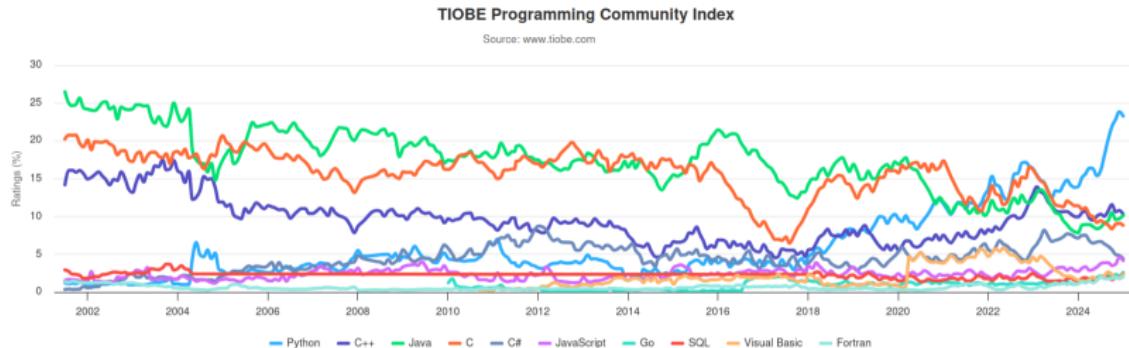
- **Arrays (Vetores):** Coleção de elementos do mesmo tipo.
- **Listas:** Sequência de elementos que podem ser adicionados ou removidos.
- **Pilhas (Stacks):** Estrutura "LIFO"(Last In, First Out).
- **Filas (Queues):** Estrutura "FIFO"(First In, First Out).
- **Árvores:** Hierarquia de dados com nós e ramificações.
- **Grafos:** Conjunto de nós conectados por arestas.

A escolha da estrutura de dados adequada é crucial para otimizar o desempenho de algoritmos e sistemas.

História das Linguagens de Programação

História das Linguagens de Programação

- Timeline das linguagens de programação
- <https://www.youtube.com/watch?v=ZTPrbAKmcdo>



Tiobe indexador [Fonte: <https://www.tiobe.com/tiobe-index/>]

Linguagem Compilada vs. Interpretada

Linguagem Compilada vs. Interpretada

- Linguagens de programação podem ser classificadas de acordo com a forma como são executadas.
- Duas abordagens principais:
 - Linguagens Compiladas
 - Linguagens Interpretadas
- Cada abordagem tem vantagens e desvantagens.

Linguagens Compiladas

- **Definição:** O código-fonte é traduzido para código de máquina (binário) antes da execução.
- **Exemplos:** C, C++, Rust, Go.
- **Vantagens:**
 - Performance superior, pois o código é otimizado para o hardware.
 - Execução direta pelo processador.
- **Desvantagens:**
 - Processo de compilação pode ser demorado.
 - Menos portável, pois o binário é específico para uma plataforma.

Linguagens Interpretadas

- **Definição:** O código-fonte é executado linha por linha por um interpretador em tempo de execução.
- **Exemplos:** Python, JavaScript, Ruby, PHP.
- **Vantagens:**
 - Portabilidade: o mesmo código pode rodar em diferentes plataformas.
 - Desenvolvimento mais rápido, sem necessidade de compilação.
- **Desvantagens:**
 - Performance geralmente inferior ao código compilado.
 - Dependência de um interpretador para execução.

Comparação

- **Linguagens Compiladas:**

- Execução mais rápida.
- Exemplo: C++ (compilada).

- **Linguagens Interpretadas:**

- Facilidade de uso e portabilidade.
- Exemplo: Python (interpretada).

- **Observação:** Algumas linguagens usam abordagens híbridas (ex: Java, que compila para bytecode e é interpretado pela JVM).

Considerações sobre Linguagem Compilada vs. Interpretada

- A escolha entre linguagens compiladas e interpretadas depende do projeto.
- Linguagens compiladas são ideais para aplicações de alta performance.
- Linguagens interpretadas são excelentes para prototipagem rápida e portabilidade.
- Ambos os tipos têm seu lugar no desenvolvimento de software.

Que Linguagem é o Java?

Que Linguagem é o Java?

- Java é uma linguagem de programação de alto nível.
- Criada pela Sun Microsystems em 1995 (atualmente mantida pela Oracle).
- Conhecida por sua portabilidade e uso em aplicações empresariais, web e mobile.

Características do Java

- **Orientada a Objetos:** Tudo em Java é um objeto (exceto tipos primitivos).
- **Portável:** "Write once, run anywhere"(escreva uma vez, execute em qualquer lugar).
- **Compilada e Interpretada:**
 - O código-fonte é compilado para bytecode.
 - O bytecode é interpretado pela JVM (Java Virtual Machine).
- **Gerenciamento de Memória:** Coleta de lixo (garbage collection) automática.

Como o Java Funciona?

- **Compilação:**

- O código-fonte (.java) é compilado para bytecode (.class) pelo compilador Java (javac).

- **Execução:**

- O bytecode é executado pela JVM, que interpreta e traduz o código para a máquina específica.

- **Portabilidade:**

- A JVM é específica para cada sistema operacional, mas o bytecode é universal.

Vantagens do Java

- **Portabilidade:** Funciona em qualquer sistema com uma JVM.
- **Segurança:** Gerenciamento de memória e verificação de bytecode.
- **Grande Ecossistema:** Bibliotecas robustas e frameworks populares (Spring, Hibernate, etc.).
- **Comunidade Ativa:** Suporte e atualizações constantes.

Desvantagens do Java

- **Performance:** Pode ser mais lento que linguagens totalmente compiladas (ex: C++).
- **Verbosidade:** Código mais longo em comparação com linguagens como Python.
- **Complexidade:** Configuração inicial pode ser desafiadora para iniciantes.

Comentários

- Java é uma linguagem poderosa e versátil.
- Combina compilação e interpretação para oferecer portabilidade e segurança.
- Amplamente utilizada em aplicações empresariais, Android e sistemas distribuídos.
- Continua sendo uma das linguagens mais populares do mundo.

VS Code Online Configuração

1. Criar um Repositório no GitHub

- Acesse <https://github.com> e faça login.
- Clique em **New** para criar um novo repositório.
- Dê um nome ao repositório (por exemplo, `java-codespaces`).
- Escolha se será público ou privado.
- Marque a opção **Add a README file** (opcional, mas útil).
- Clique em **Create repository**.

2. Configurar o Codespace

- No repositório criado, clique na aba **Code** e selecione **Codespaces**.
- Clique em **Create codespace on main**.
- O GitHub vai criar um ambiente de desenvolvimento baseado em contêineres.
- Isso pode levar alguns minutos na primeira vez.

3. Configurar o Ambiente para Java

Crie uma pasta e adicione o arquivo `.devcontainer/devcontainer.json` com o seguinte conteúdo:

```
{  
  "name": "Java Development",  
  "image": "mcr.microsoft.com/devcontainers/java:17",  
  "features": {  
    "ghcr.io/devcontainers/features/java:1": {  
      "version": "17",  
      "installMaven": true,  
      "installGradle": true  
    }  
  },  
  "customizations": {  
    "vscode": {  
      "extensions": [  
        "vscjava.vscode-java-pack"  
      ]  
    }  
  }  
}
```

4. Continuação e Verificação a Instalação do Java

O conteúdo esta no notebook da aula:

Estrutura_de_dados_aula_1_v2.ipynb

- Salve o arquivo e reinicie o Codespace.

No terminal do Codespace, execute:

```
java --version  
mvn --version  
gradle --version
```

- Verifique se as saídas mostram as versões corretas.

5. Criar e Executar um Projeto Java

Crie um arquivo `HelloWorld.java`:

O conteúdo esta no notebook da aula:

`Estrutura_de_dados_aula_1_v2.ipynb`

Compile e execute (um exemplo está no notebook da aula).

Considerações

Considerações

■ Recordando!!!

- Programação do curso.
- Estrutura de dados.
- História das linguagens de programação.
- Estrutura de dados com Java e Python!
- Tipos dados com Java e Python.
- Exemplos e Exercícios.