

~~MAIS UM CURSO DE~~ JAVA E PROGRAMAÇÃO ORIENTADA A OBJETOS ~~QUE NINGUÉM PEDIU~~

COM INTRODUÇÃO À ENGINE DE JOGOS JSGE
E EXERCÍCIOS CRIATIVOS

INTRODUÇÃO

PROF. DR. DAVID BUZATTO

INTRODUÇÃO

- Objetivo

- Tentar ensinar, quem não sabe programar, o básico sobre algoritmos e programação de computadores, usando a linguagem de programação Java;

- Motivação

- Necessidade de criar um curso de verdade, **GRATUITO** e que preste sobre o tema;

- Público alvo

- Qualquer pessoa que tenha e que saiba usar um computador pessoal, que goste de ciências exatas e que queira aprender o bô-á-bá sobre programação, sem conversa fiada!

REQUISITOS

- Um computador pessoal, não muito antigo;
 - Não vai dar certo fazer pelo celular, nem pelo tablet e nem em máquinas muito ultrapassadas;
- Windows 10 ou superior;
 - Quem usa Linux pode fazer também, mas vai ter que se virar com ferramentas e configurações;
- Vontade de aprender;
- Disponibilidade para treinar;
 - Teremos MUITOS exercícios;

O QUE SERÁ USADO

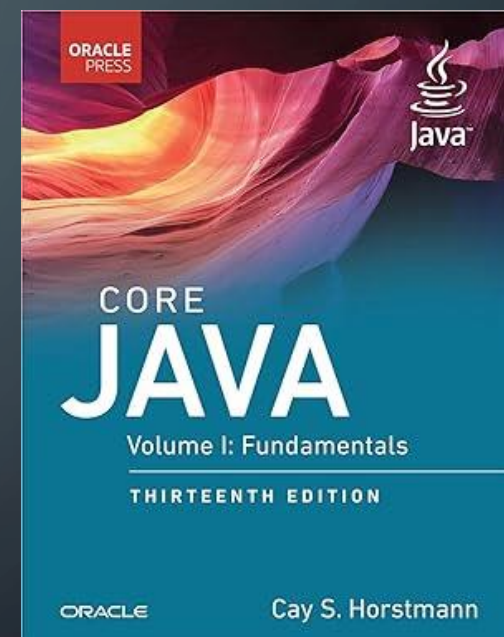
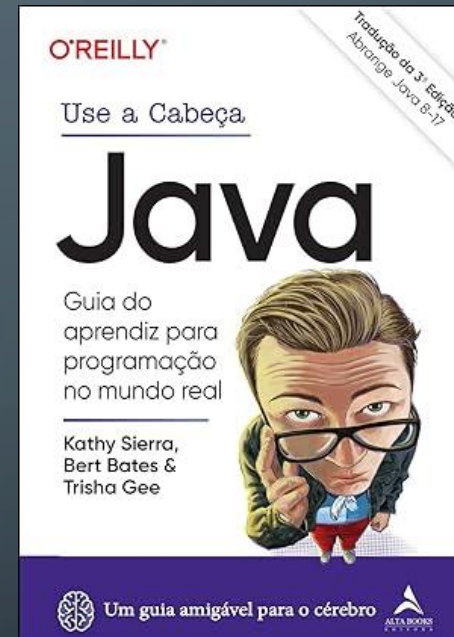
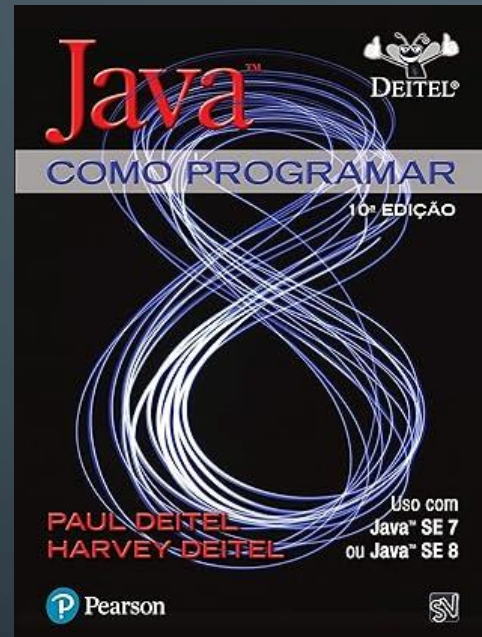
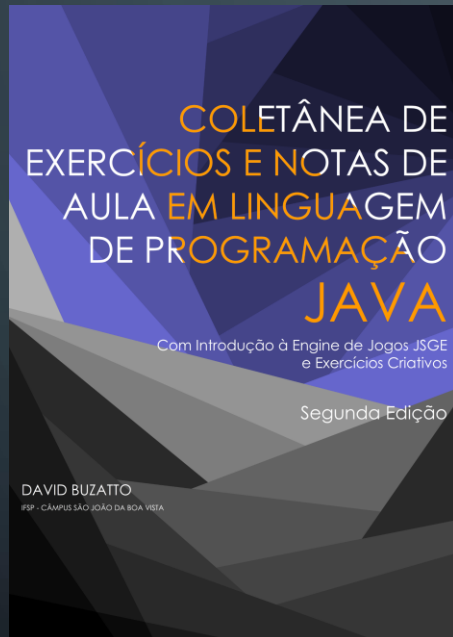
- Linguagem de programação Java;
 - Usaremos o JDK da Oracle;
- Visual Studio Code;
 - Usaremos algumas extensões para nos ajudar;
- Apache NetBeans;
 - Usaremos para projetos maiores;
- JSGE;
 - Engine (motor) para desenvolvimento de jogos;
- JJudge;
 - Corretor de exercícios;
- Livro: “Coletânea de Exercícios e Notas de Aula em Linguagem de Programação Java: com introdução à engine de jogos JSGE e exercícios criativos”, 2ª edição.

CONTEÚDO

1. Preparação do ambiente;
2. Hello World!
3. Entrada e saída padrão formatados;
4. JSimple Game Engine (JSGE);
5. Estruturas condicionais;
6. Estruturas de repetição;
7. Arrays (~~não são vetores pô!~~);
8. Classe Math;
9. Métodos estáticos;
10. Algoritmos de ordenação elementares;
11. Caracteres e Strings;
12. Arquivos;
13. Recursividade;
14. Algoritmos de ordenação não elementares;
13. Classes, atributos e métodos;
14. Encapsulamento;
15. Composição, agregação e associação;
16. Herança e polimorfismo;
17. Interface gráfica com Swing;
18. Interfaces;
19. Enumerações;
20. Records (registros);
21. Coleções (*Java Collections Framework*);
22. Expressões lambda, interfaces funcionais e Stream API;
23. Metaprogramação;
24. Resolução de exercícios por capítulo do livro;
25. Desenvolvimento de simulações e jogos usando JSGE.

BIBLIOGRAFIA SUGERIDA

6/17



VAMOS COMEÇAR?

O QUE É ALGORITMO?

- **Informal:**

- Conjunto finito de instruções, executadas em ordem lógica, que tem como objetivo resolver um problema;

- **Formal:**

- Conjunto das regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número de etapas (Dicionário Houaiss da Língua Portuguesa, 2001);
- *The term algorithm is used in computer science to describe a problem-solving method suitable for implementation as a computer program (Algorithms in C, Sedgewick, 1998);*
- *Methods for solving problems that are suited for computer implementation (Algorithms, Sedgewick, 2011).*

PENSAR COMO UM CIENTISTA DA COMPUTAÇÃO

- O que é computação?
- Tipos de conhecimento:
 - **Declarativo:** definição de algo;
 - $\sqrt{x} = y \Rightarrow y^2 = x$ e $y \geq 0$
 - **Imperativo:** como fazer algo (receita);

```
início
  escolher um valor para y
  se  $y^2 \approx x$  então
    pare  $\rightarrow y$ 
  senão
     $y \leftarrow \frac{(y + \frac{x}{y})}{2}$ 
  repita
```

PENSAR COMO UM CIENTISTA DA COMPUTAÇÃO

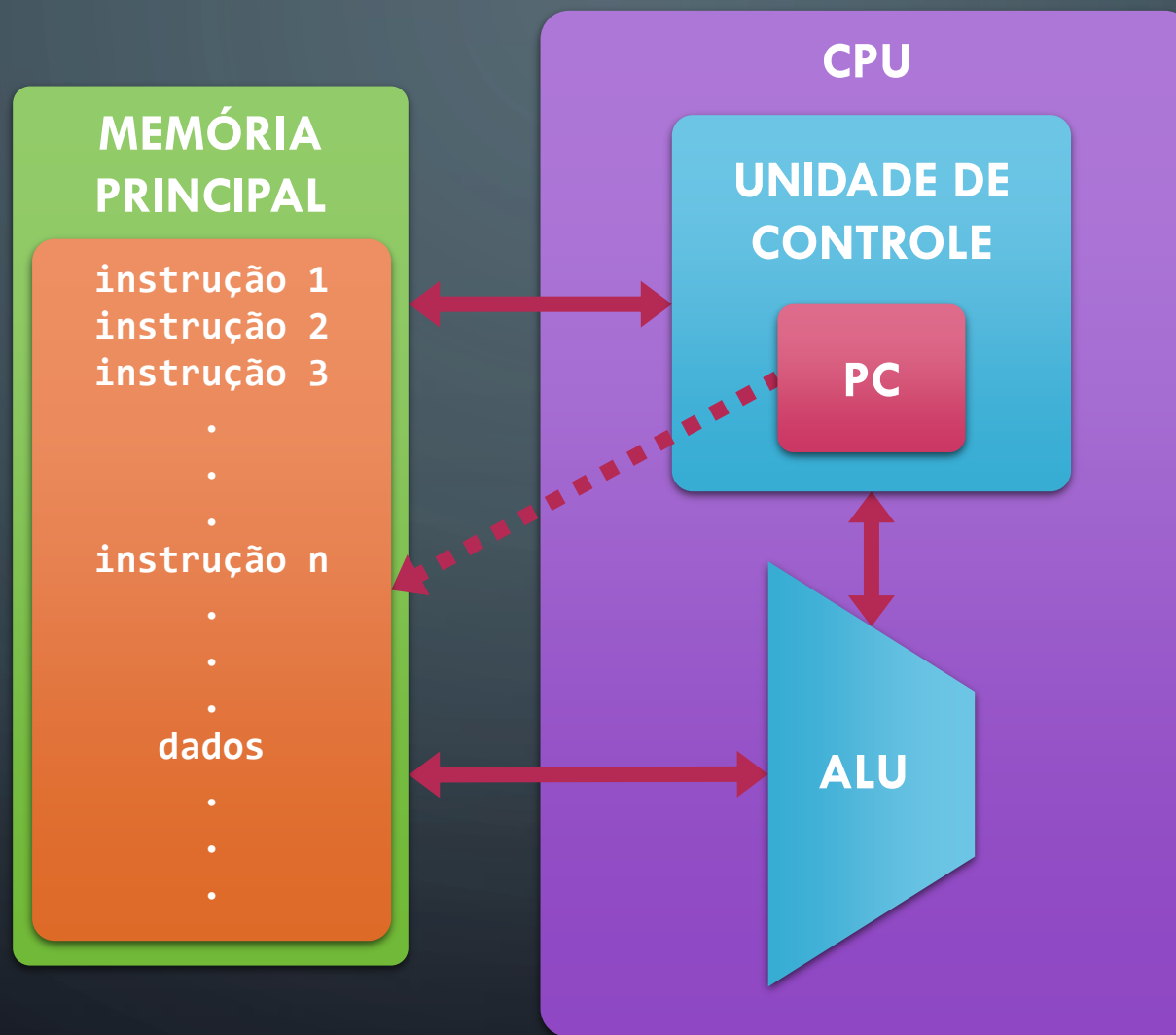
- Como automatizar “mecanicamente” a receita apresentada?
 - **Computador de programa fixo:**
 - Conjunto de circuitos projetado para resolver apenas um tipo problema;
 - **Exemplo:** calculadora.

PENSAR COMO UM CIENTISTA DA COMPUTAÇÃO

- Ao invés de haver uma máquina com um circuito fixo, que tal haver uma máquina que recebe como entrada um circuito, ou uma receita, se reorganiza internamente, e é capaz de realizar aquela receita, ou seja, que tenha a capacidade de agir como a receita?
- **Computador de programa armazenado!**

ARQUITETURA DE VON NEUMANN

12/17



TIPOS DE LINGUAGENS DE PROGRAMAÇÃO

13/17

Alto Nível



Linguagem Natural



Linguagem de Máquina

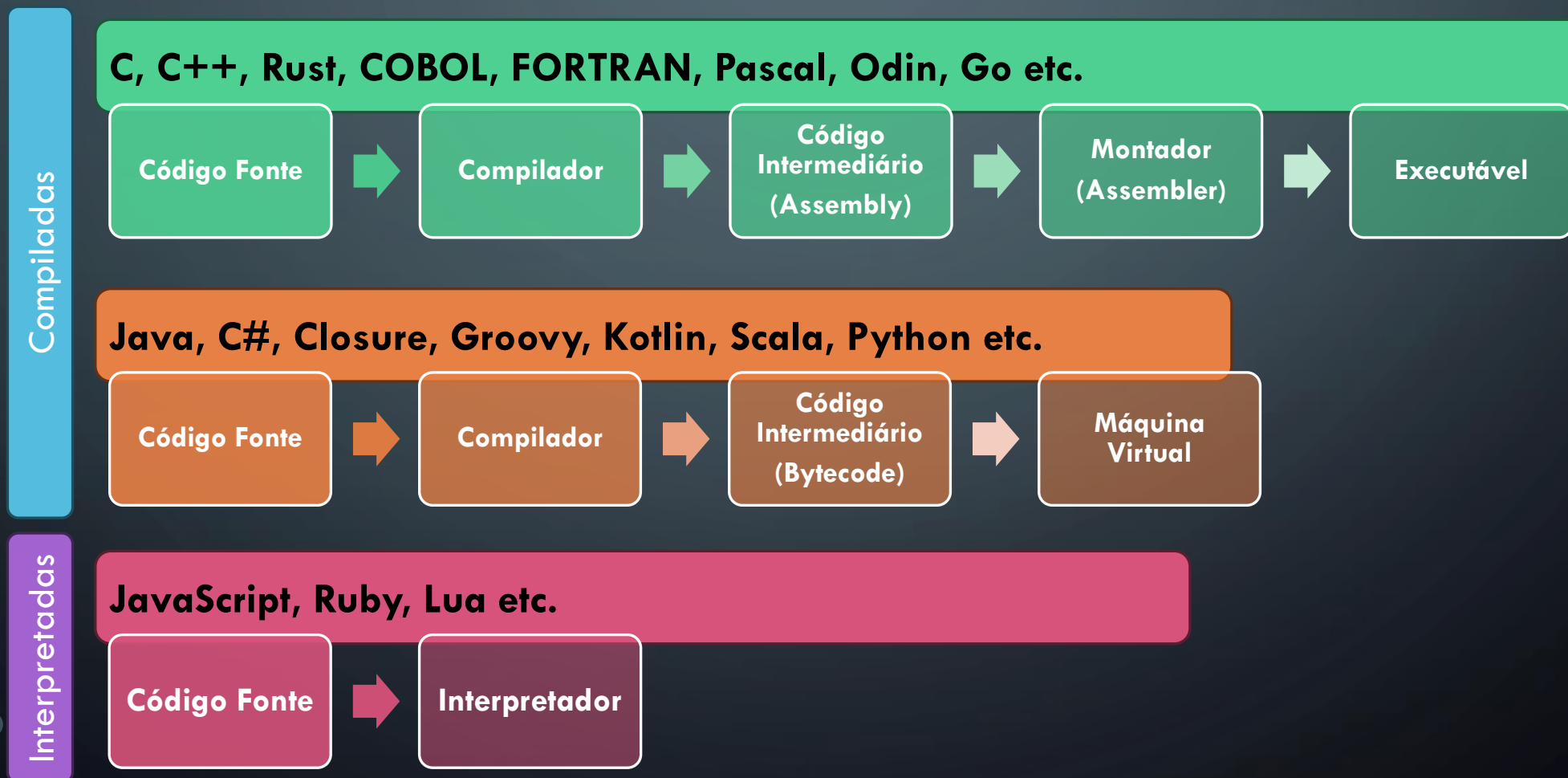
Baixo Nível

LINGUAGENS DE PROGRAMAÇÃO

- Propósito geral ou específica de domínio:
 - **Geral:** linguagem de programação projetada para resolver problemas de diversas áreas do conhecimento ou para realizar atividades diversas;
 - C, C++, **Java**, Python, Rust, Pascal, Ruby, Odin, Zig, Go etc.
 - **Específica de domínio:** linguagem de programação projetada para resolver problemas e/ou realizar atividades de áreas específicas.
 - HTML (*Hypertext Markup Language*);
 - SQL (*Structured Query Language*);
 - HDL (*Hardware Definition Language*);
 - MATLAB (MATrix LABoratory) etc.

LINGUAGENS DE PROGRAMAÇÃO

- Compilada ou Interpreta:



LINGUAGEM DE PROGRAMAÇÃO JAVA

- Linguagem de programação orientada a objetos, de alto nível e de propósito geral, compilada em código intermediário (bytecode) que é interpretado por uma máquina virtual, a Java Virtual Machine (JVM);
- Lançada em 1995 e desenvolvida inicialmente por James Gosling (e outros) na Sun Microsystems;
- Atualmente é propriedade da Oracle Corporation;
- Padrão *de facto* controlado pelo Java Community Process (JCP);
- Atualizada constantemente;
- Última versão: Java 23;
- Novas versões principais (*major versions*) a cada 6 meses (aproximadamente);
- A implementação de referência é o OpenJDK;
- Ecossistema GIGANTESCO.

MÃOS À OBRA!