

# Conceitos básicos

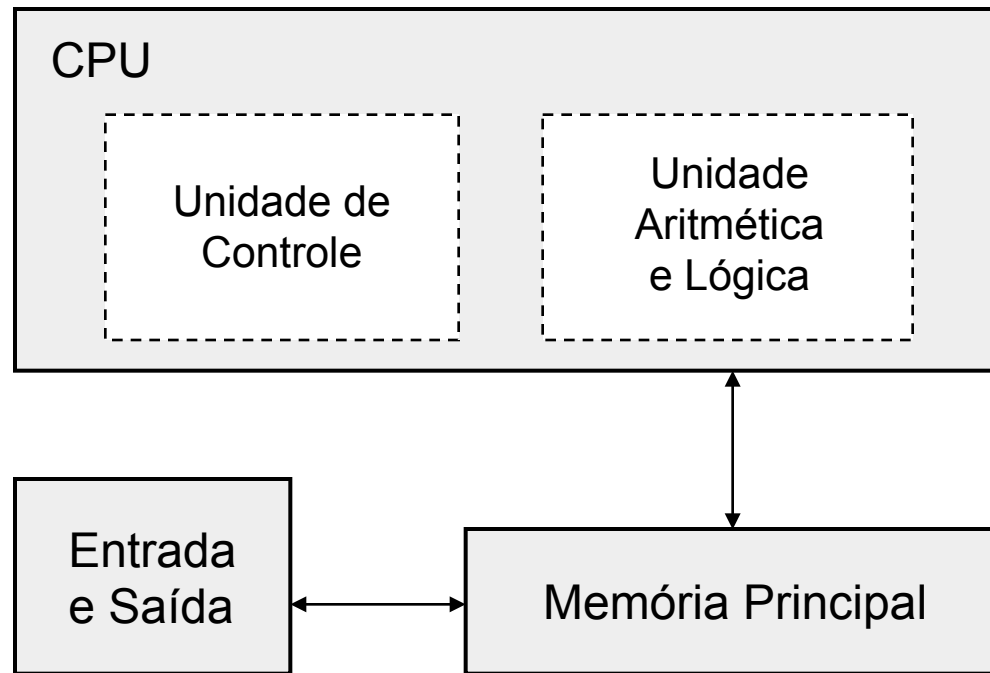
Mateus Raeder

# Introdução

## Arquitetura de Von Neumann

- ▶ Apesar das mudanças na informática nas últimas décadas, grande parte dos computadores ainda segue a arquitetura de Von Neumann
- ▶ Esta arquitetura foi proposta por Neumann (John Von Neumann) na década de 40
- ▶ Composto por basicamente 3 subsistemas: **CPU, Unidade de Memória Principal (RAM) e Unidades de Entrada/Saída**

# Arquitetura de Von Neumann



*Arquitetura de Von Neumann*

# Arquitetura de Von Neumann

→ Mas o que é uma instrução?

- **Unidade de Controle:** identifica e decodifica as instruções, acionando a unidade responsável pela sua execução
- **Unidade Aritmética e Lógica:** executa as operações lógicas e aritméticas
- **Unidade de Memória Principal (RAM - Random Access Memory):** armazena o programa e seus respectivos dados. O programa e os dados devem estar na memória RAM no momento da execução.
- **Unidades de Entrada:** transferem informações externas à máquina para a Unidade de Memória Principal (por exemplo, mouse, scanner, teclado, webcam, microfone)
- **Unidades de Saída:** exibem as informações que encontram-se guardadas na Unidade de Memória Principal através de algum meio que seja legível aos seres humanos (por exemplo, monitor de vídeo, caixas de som, impressora)

# Instruções

## Mas o que é uma instrução?

- Instrução (ou comando) é uma regra que vai indicar ao computador quais os dados a serem operados e qual a operação a ser realizada.

- Por exemplo: 

|   |   |   |   |
|---|---|---|---|
| 7 | – | 5 | = |
|---|---|---|---|

- A CPU, então, executa alguns passos para que cada uma das instruções sejam corretamente efetuadas

# Instruções - etapas de processamento

**As etapas para o processamento de uma instrução são (basicamente):**

- A Unidade de Controle busca a instrução na memória
- A Unidade de Controle decodifica a instrução, definindo as posições de memória necessárias para sua execução
- A Unidade de Controle transfere os dados para a Unidade Aritmética e Lógica
- A Unidade Aritmética e Lógica executa as operações sobre os dados e o resultado é armazenado na memória

# Linguagem de Máquina x Linguagem de Alto Nível

- Podemos, então, passar para a máquina o conjunto de instruções que desejamos que sejam executadas
- Porém, a máquina não entende a nossa linguagem
- Então, como passamos estas instruções para a máquina?

Linguagem de máquina

Tipo mais primitivo de linguagem. É a linguagem que o computador entende.

Linguagem de programação de alto nível

Tipo de linguagem criada para facilitar a comunicação com a máquina. Deve ser traduzida em linguagem de máquina.

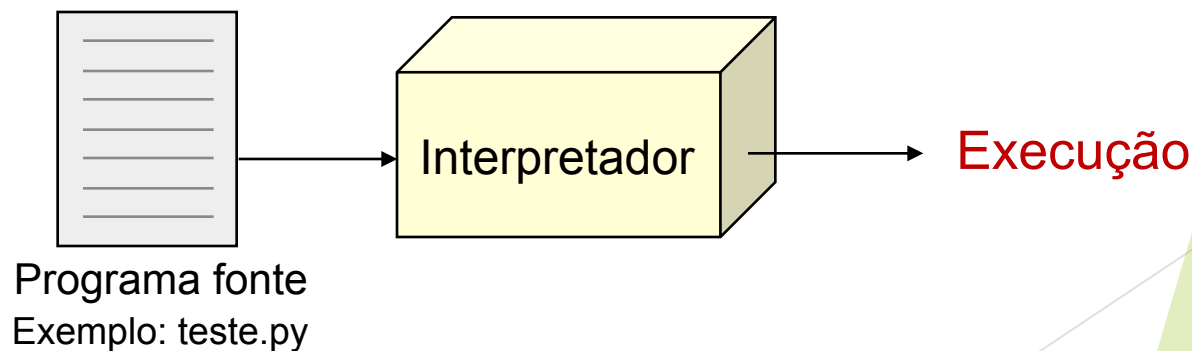
# Processos de tradução

- Existem diversas linguagens de programação de alto nível. Por exemplo: C, C++, Java, Pascal, Delphi, etc.
- Os programas de computador são, então, escritos utilizando uma linguagem de programação de alto nível (na qual um código ou programa fonte é criado) e são traduzidos para a linguagem de máquina
- Os 2 principais processos que traduzem programas escritos em uma linguagem de alto nível para a linguagem de máquina são **interpretação** e **compilação**



# Processo de Interpretação

- Um interpretador é um programa que recebe como entrada um arquivo contendo um programa fonte
- O interpretador lê linha a linha este arquivo de entrada, e executa **uma a uma** as instruções que estão nele programadas



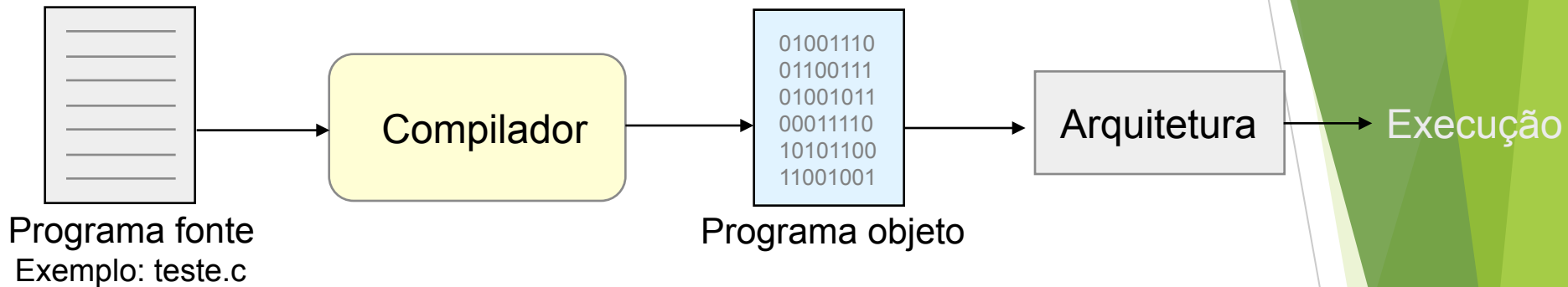
# Processo de Interpretação

- Um programa pode ser executado em qualquer máquina, desde que haja o interpretador disponível para utilização
- Porém, pode-se perceber que a execução de um programa que utiliza um interpretador é relativamente lenta, pois a cada nova linha lida no programa fonte, esta deve ser decodificada antes de ser executada
- Se houver algum erro de no código (que vá contra as definições da linguagem), o programa será interrompido abruptamente com um erro

# Processo de Compilação

- Um compilador é um programa (ou um conjunto de programas) que recebe como entrada um programa fonte e cria um novo arquivo como saída
- O arquivo de entrada é o código fonte escrito pelo programador
- O arquivo que o compilador produz, é normalmente identificado como código objeto
- Este código objeto contém instruções de baixo nível, traduzidas para a linguagem de máquina

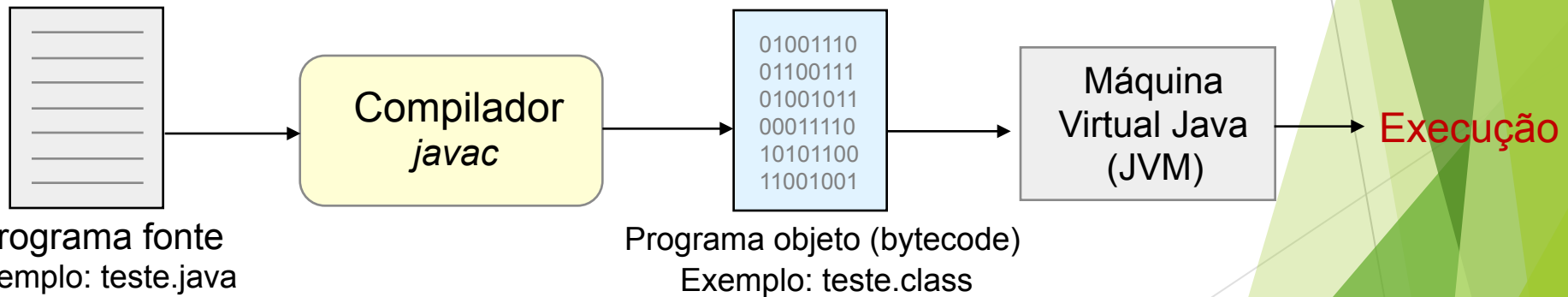
# Processo de Compilação



- O código produzido não é portátil para qualquer arquitetura
- Diferentes compiladores são construídos para as diferentes arquiteturas de processadores (diferentes famílias de processadores possuem conjuntos diferentes de instruções)
- O compilador somente gera corretamente o código objeto caso não tenha encontrado um **erro de compilação**

# Linguagem Java

- Java é uma linguagem compilada
- O compilador Java, entretanto, não gera código para um processador específico, mas sim para uma máquina virtual Java (a JVM - Java Virtual Machine)



# Linguagem Java

- Os programas escritos em Java são, portanto, portáveis sobre todos os computadores para os quais existir uma implementação da JVM

- Desta forma, Java possui tanto benefícios característicos das linguagens **compiladas**

*uma vez que a execução de um programa não corre o risco de ser abortada devido a um erro de sintaxe no seu código*

- quanto das linguagens **interpretadas**

*uma vez que executa sobre uma máquina virtual que tem um comportamento semelhante a um interpretador*