



BandTec

DIGITAL SCHOOL



LP

Linguagens de Programação

Aula 1 – Conceitos Básicos

São Paulo - 2020

- As linguagens de programação podem ser de:
 - Baixo nível – mais próximas do hardware, as instruções são mais simples, a programação e o entendimento de um programa é mais difícil
 - Alto nível – não são tão dependentes do hardware, as instruções são mais complexas, facilitando a programação e o entendimento do programa

- Linguagem de máquina
 - Conjunto de instruções de um processador
 - Expresso através de zeros e uns
 - O processador e os circuitos eletrônicos internos do computador só “entendem” os zeros e uns (notação binária)
 - Zero – indica “desligado”
 - Um – indica “ligado”

Linguagem de baixo nível: Assembly

- Linguagem Assembly (ou linguagem de montagem)
 - Instruções da linguagem de máquina em binário passam a ser expressos através de um mnemônico (sigla para facilitar a programação)
 - Passa a existir o conceito de variável, que é um nome associado a uma determinada posição da memória
 - Exemplo de código (acumulador é uma memória interna ao processador):
LDA 5 ;carregue o valor 5 ao acumulador
ADD 3 ;some 3 ao valor do acumulador
STA X ;armazene o valor do acumulador na variável X

Montador ou Assembler

- Montador ou Assembler
 - Software que traduz o programa escrito em Assembly para o programa equivalente em linguagem de máquina



Linguagem de alto nível

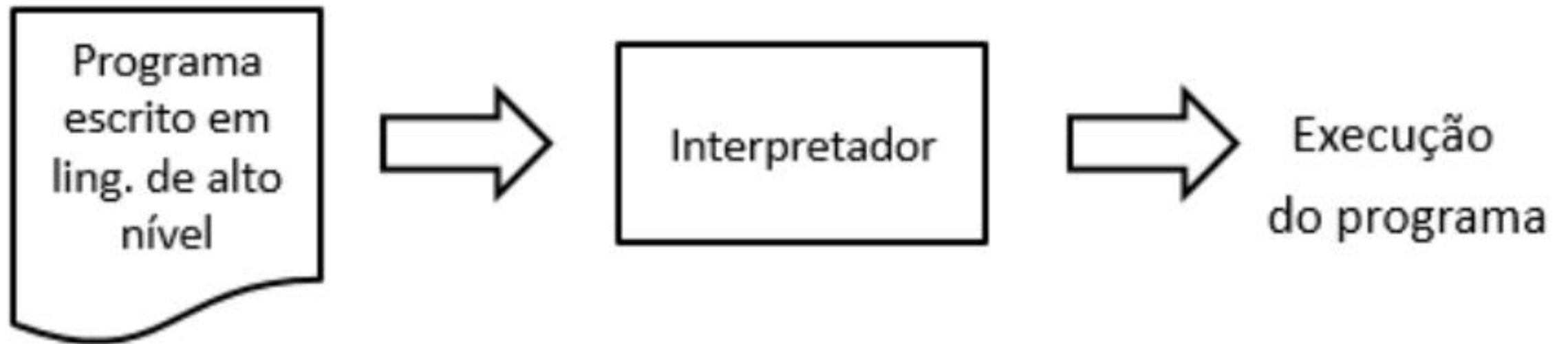
- Linguagem de baixo nível – desvantagens:
 - Dependente do processador
 - Um programa escrito para o processador A precisa ser reescrito para poder ser executado em outro processador B
 - Programação e entendimento do programa é complexa
- Linguagem de alto nível
 - Criada para resolver os problemas acima
 - Exemplo: Fortran, Cobol, Pascal, Basic, C, C++, Java, C#, PHP, Python, Ruby, etc.
 - Instruções são mais próximas da linguagem humana
 - Não está associada ao processador

- **Compilador**
 - Software que traduz o programa escrito em linguagem de alto nível para o programa equivalente em linguagem de baixo nível



- **Compilador**
 - Analisa o programa escrito em linguagem de alto nível, verificando se não há erros sintáticos, como a falta de caracteres, como ')', ';', etc, e se não há instrução escrita de forma errada
 - Se não tiver erro sintático, gera o programa equivalente em linguagem de baixo nível
 - Uma vez gerado o código de baixo nível, pode-se executá-lo várias vezes, sem a necessidade de compilá-lo novamente
 - O código gerado depende do processador e do sistema operacional da máquina para a qual foi gerado
 - Exemplo de linguagens compiladas: C, C++, Fortran

- Interpretador
 - Software que interpreta e executa o programa escrito em linguagem de alto nível, sem gerar o programa equivalente em linguagem de baixo nível

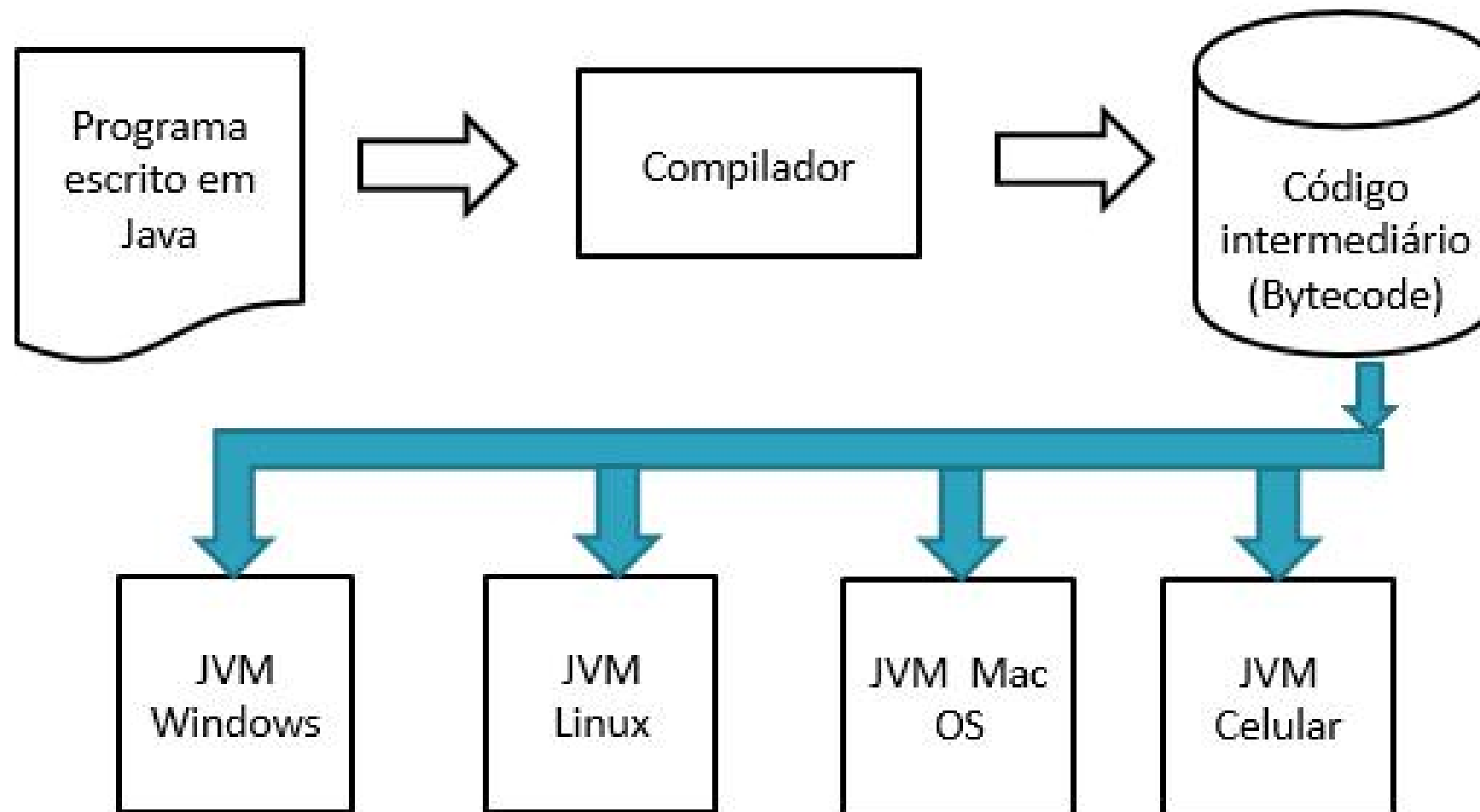


- Interpretador
 - Também analisa o programa com relação a erro sintático
 - Se não tiver erro, executa o programa
 - Toda vez que o programa for executado, será feita a análise sintática, o que torna a execução mais demorada do que a execução do código compilado
 - É mais flexível com relação a plataformas diferentes, desde que exista o interpretador para várias plataformas
 - Exemplo de linguagens interpretadas: Basic, Prolog, PHP

Linguagens híbridas - Ex: Java

- Programa escrito em linguagem C ou C++
 - Para ser executado no Windows, precisa ser compilado por um compilador para Windows
 - Para ser executado no Linux, precisa ser compilado por um compilador para Linux
 - O início do programa deve ser alterado dependendo do Sistema Operacional (diretivas de uso de bibliotecas diferentes)
- Programa escrito em linguagem Java (híbrida)
 - Passa por 2 fases:
 - Compilação – geração de um código intermediário (Bytecode)
 - Interpretação – bytecode é interpretado e executado pelo JVM (Java Virtual Machine)

- Slogan do Java: “Write once, run anywhere”



- IDE – Integrated Development Environment
 - Ambiente de Desenvolvimento Integrado
 - Software muito utilizado atualmente para desenvolvimento de programas
 - Contém editor de texto (para editar os programas em várias linguagens), compilador, ambiente para executar os programas, depurador (para procurar erros durante a execução do programa)
 - Exemplo: Visual Studio, Eclipse, NetBeans, IntelliJ IDEA