A LINGUAGEM JAVA*

Adair Santa Catarina
Curso de Ciência da Computação
Unioeste – Campus de Cascavel – PR

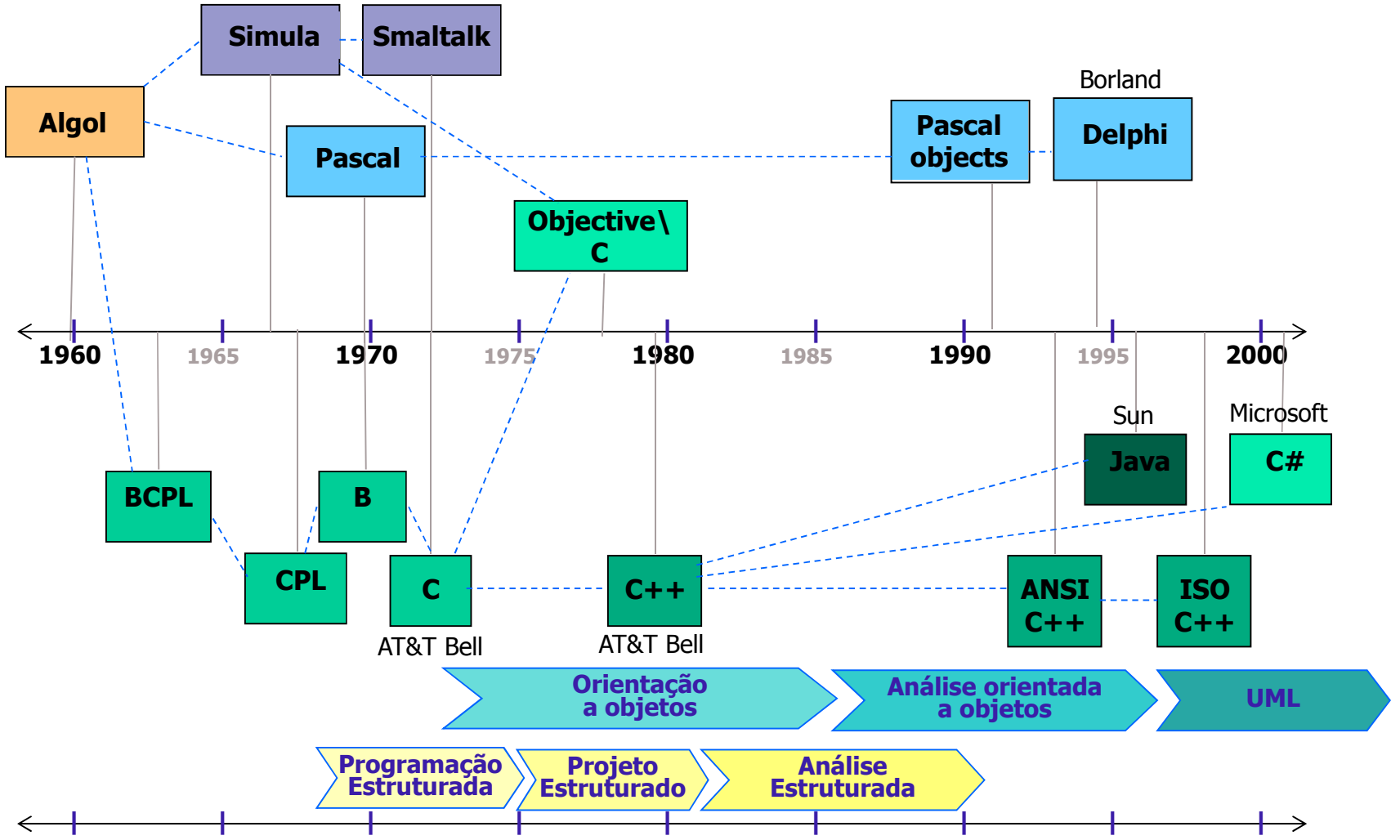
Fev/2019

*Adaptado de PACHECO, R. C. S. & RIEKE, R. N. INE – UFSC. Disponível em: http://www.stela.ufsc.br/~pacheco/dsoo/htm/downloads.htm

Objetivo das linguagens de programação

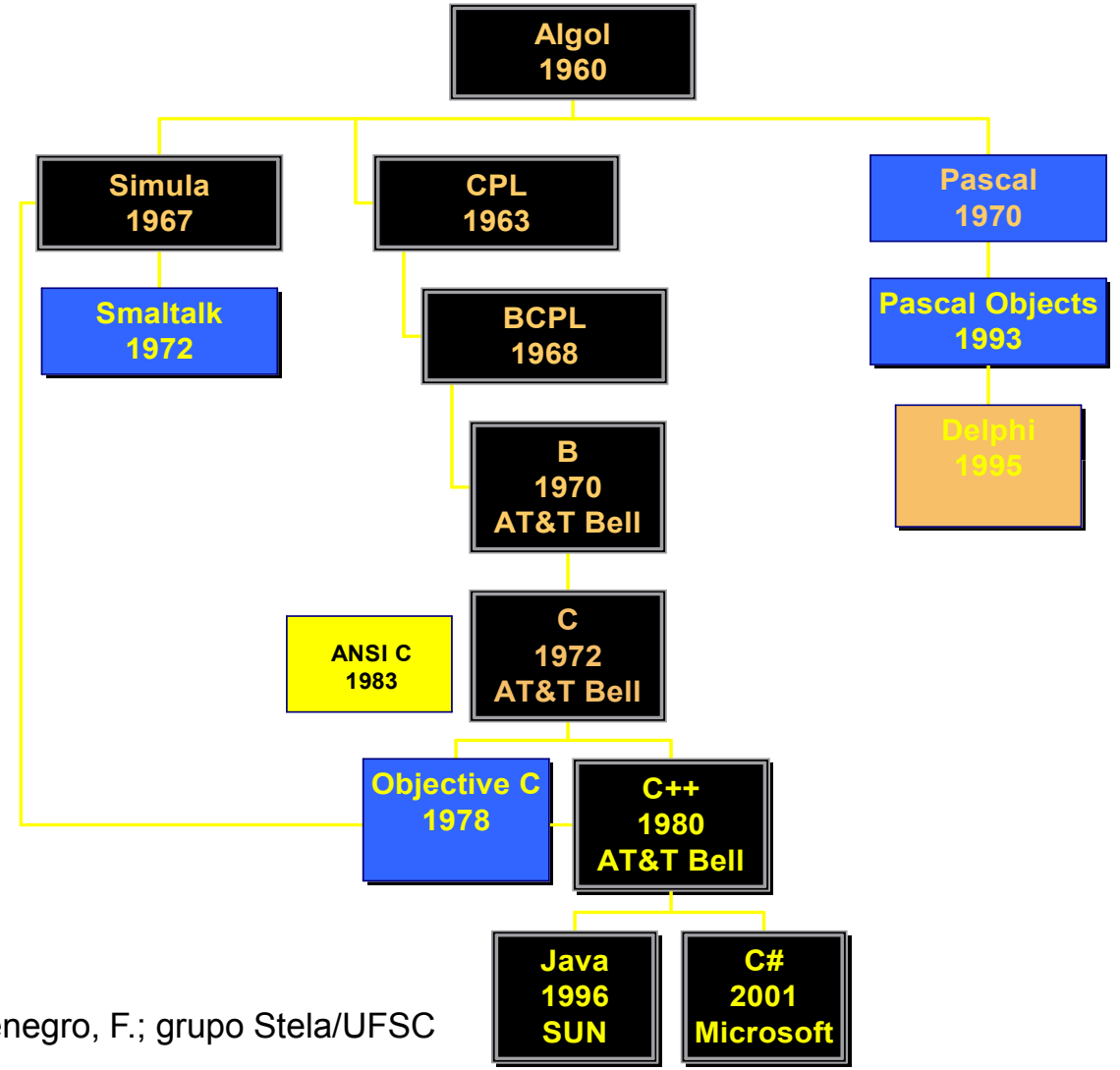
- As linguagens de programação são, em sua maioria, criadas para tratar problemas sob determinado enfoque:
  - FORTRAN → Científica;
  - COBOL → Economia;
  - PROLOG, LISP → Inteligência artificial;
  - CLIPPER → Banco de dados.
- C e C++: para todas as áreas;
  - Em especial na construção de outros ambientes;
- Java: para todas as áreas
  - Em especial para Internet, sistemas em rede e computação móvel.

Árvore genealógica das linguagens



Fonte: Montenegro, F.; grupo Stela/UFSC

A família da linguagem Java



Fonte: Montenegro, F.; grupo Stela/UFSC

Breve histórico

- Projeto "Green":
  - 1991 → James Gosling e engenheiros da Sun;
  - Objetivo: criar programas para controlar produtos eletrônicos;
  - Linguagem Oak, baseada em C++.
- Tentativa frustrada:
  - Controle remoto inteligente → não obteve patrocínio.
- Influência da web:
  - 1994: HotJava utilizando as funcionalidades adicionais do Java (independência de plataforma, confiabilidade, segurança, processamento em tempo real, etc.);
  - 1995: Netscape Navigator 2.0, compatível com Java 1.0.

Princípios da linguagem Java

- Java é:
  - Mais que uma linguagem de programação;
  - Uma completa plataforma de soluções para tecnologia.
- Portabilidade de código (*Write once, run everywhere!*);
- Possibilita desenvolver programas multitarefas (*multithreading*);
- Faz verificação automática do código carregado para garantir segurança;
- Linguagem orientada a objetos;
- Não suporta herança múltipla → interface;
- Arrays "verdadeiros" → ausência de aritmética de ponteiros.

A tecnologia Java

- A tecnologia Java é composta por uma gama de produtos, baseados no poder das redes e na ideia de que um software deveria ser executado em diferentes máquinas, sistemas e dispositivos;
- Programas em Java são executados em diferentes ambientes graças à Java *Virtual Machine* (JVM);
- A tecnologia Java é subdividida em:
  - Java SE (Java Standard Edition);
  - Java EE (Java Entrerprise Edition);
  - Java ME (Java Mobile Edition);
  - Java Embedded;
  - Java Card;
  - Java TV.

A tecnologia Java

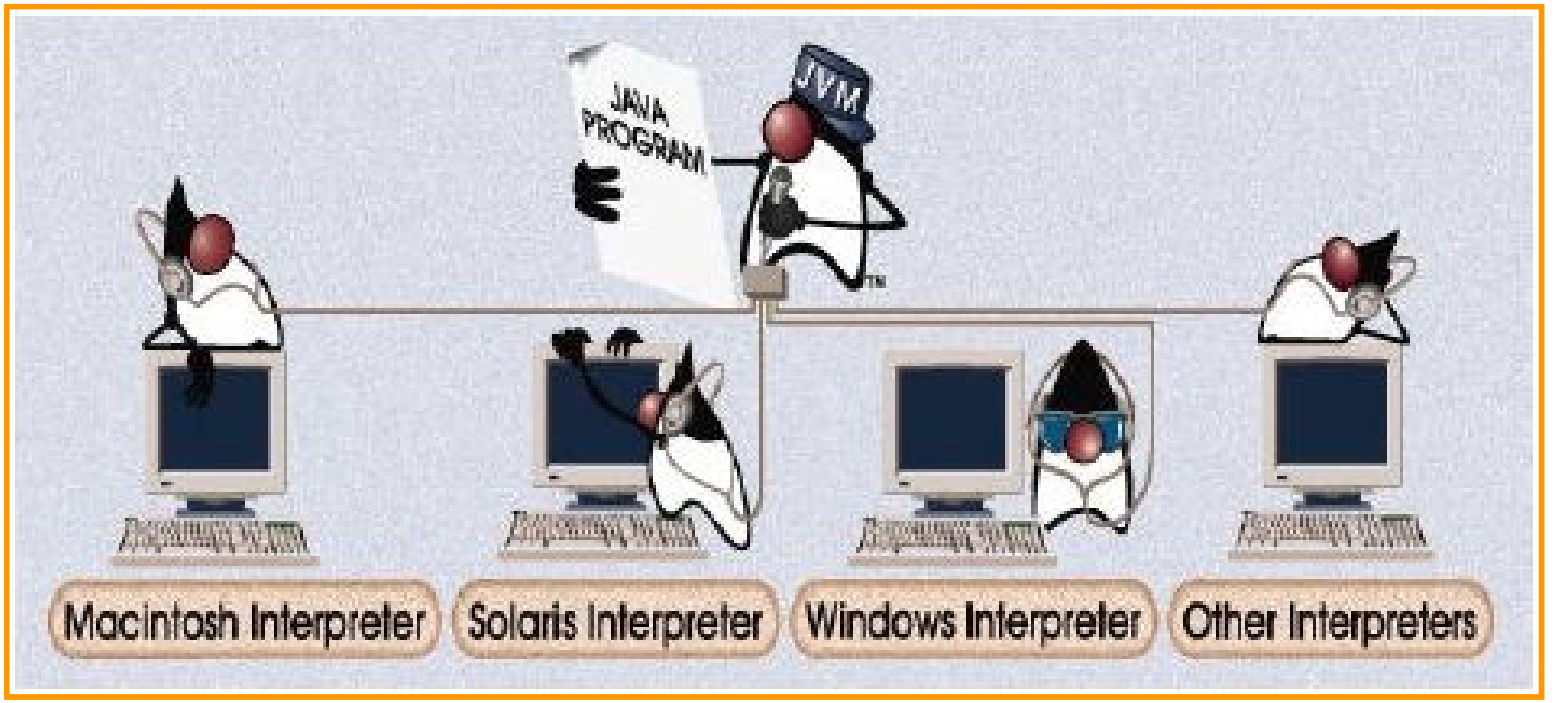
- **Java SE:**
  - Contém as classes principais da plataforma Java (*Core Java Plataform*);
  - SDK: compilador, debugger, gerador de documentação (Javadoc), empacotador (jar) e a JRE (JVM + outros componentes.
- **Java EE:**
  - Não é um produto, mas sim uma especificação definida pela Sun;
  - Simplifica o desenvolvimento de aplicações empresariais em multi-camadas (regras de negócio, interface e banco de dados);
  - Baseada em componentes padronizados, modulares e reusáveis (EJB).
- **Java ME:**
  - Dispositivos móveis: celulares, PDAs;
- **Java Embedded:**
  - Desenvolvimento de aplicativos para IoT (Internet das Coisas).
- **Java Card:**
  - Possibilita executar pequenos applets Java, com segurança, em dispositivos com processamento e armazenamento limitados.
- **Java TV:**
  - Adaptação da Java ME para desenvolver aplicativos para TVs e Receptores.

A Máquina Virtual Java (JVM)

- Uma Máquina Virtual é uma máquina imaginária que é implementada pela sua simulação em uma máquina real;
- O código para a Máquina Virtual Java está armazenado em arquivos .class, cada um contendo código para ao menos uma classe;
- A JVM interpreta byte codes (são independentes de plataforma de hardware);
- A JVM pode ser implementada em software ou hardware (computação móvel);
- Byte codes tornam a linguagem Java portável para diversas plataformas;
- Qualquer interpretador Java tem sua máquina virtual.

A Máquina Virtual Java (JVM)

- Cabe ao interpretador Java de cada plataforma de hardware específica assegurar a execução do código compilado para a JVM.



O que há na JVM?

- Conjunto de instruções (equivalente a uma CPU);
- Conjunto de registradores;
- Arquivo no formato .class;
- Pilhas;
- Coletor de lixo (Garbage Collection);
- Área de memória.

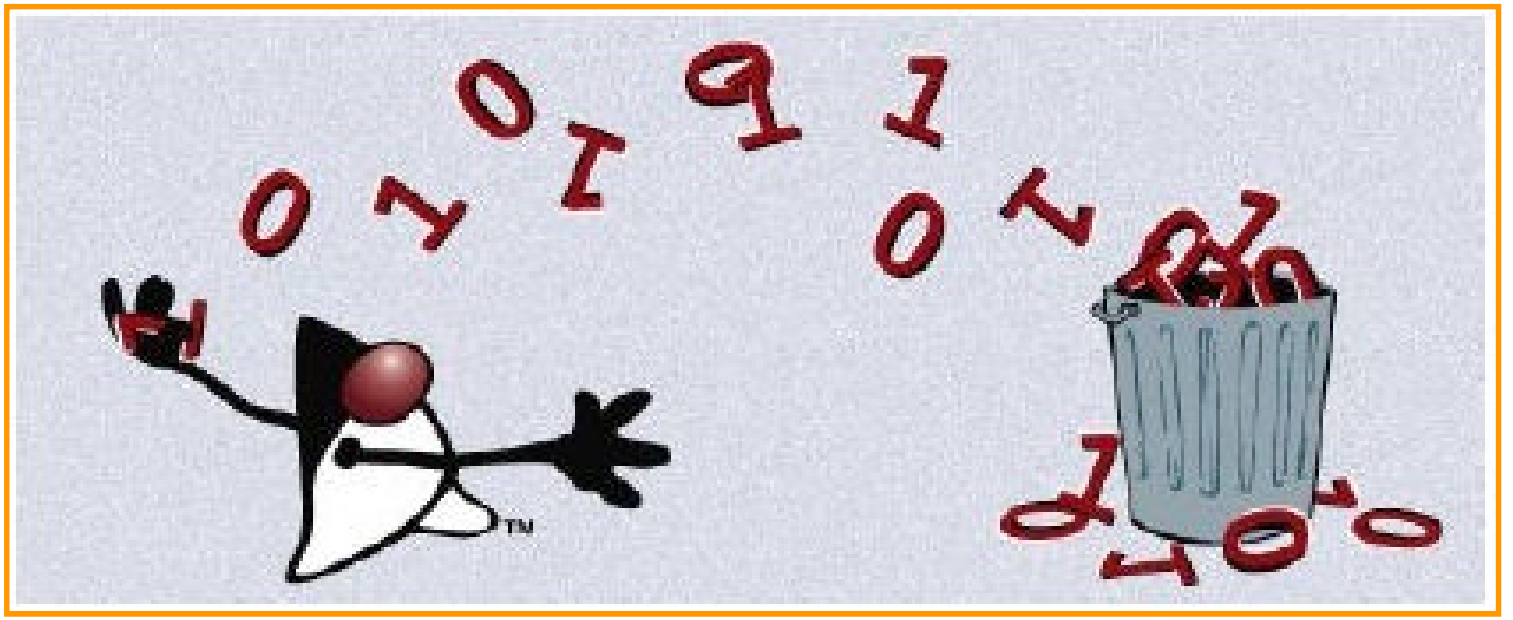
- O código executado pela JVM são byte codes compactos e eficientes. A maior parte da checagem de tipo é feita em tempo de compilação.

Garbage Collection (Coletor de Lixo)

- Nas linguagens tradicionais a liberação da memória alocada dinamicamente fica sob responsabilidade do programador;
- Java fornece um sistema a nível de thread para registrar a alocação de memória e liberar a memória que não está sendo utilizada;
- A JVM utiliza uma thread de baixa prioridade da biblioteca System (**System.gc()**);
- Ela pode também ser chamada de forma explícita pelo programador.

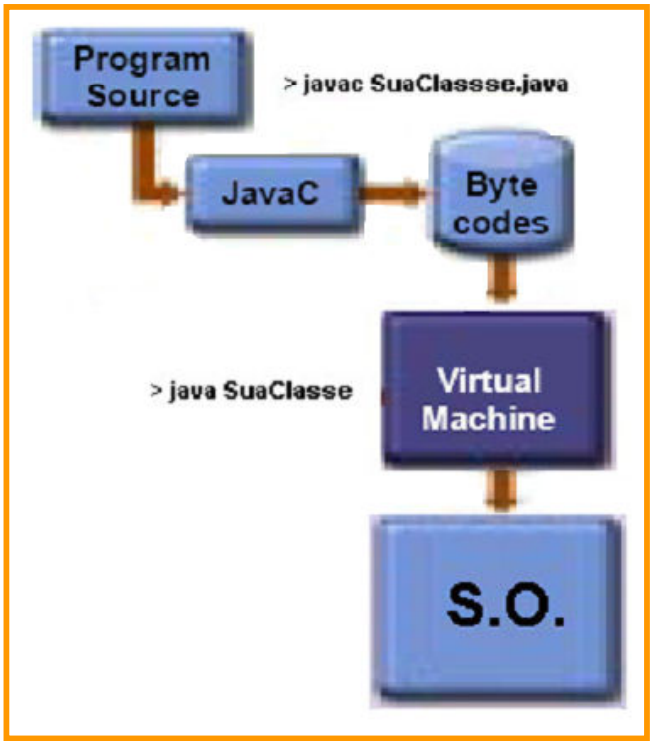
Garbage Collection (Coletor de Lixo)

- A Linguagem Java realiza a liberação de memória, que antes era responsabilidade do programador.



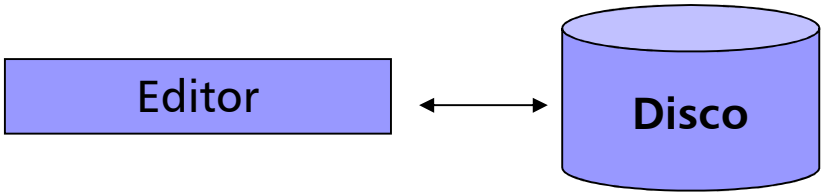
Fundamentos da linguagem Java

- Programas em Java quando compilado → código intermediário (bytecode);
- Este código é verificado e carregado na memória para então ser interpretado pela JVM;
- O Java **NÃO** gera executáveis.



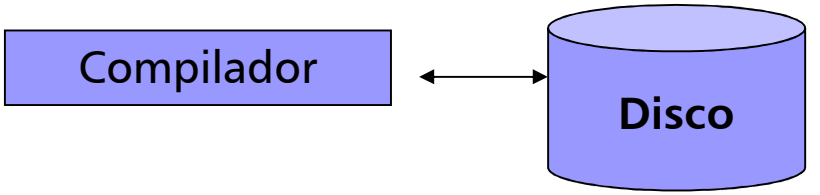
Visão geral

Fase 1



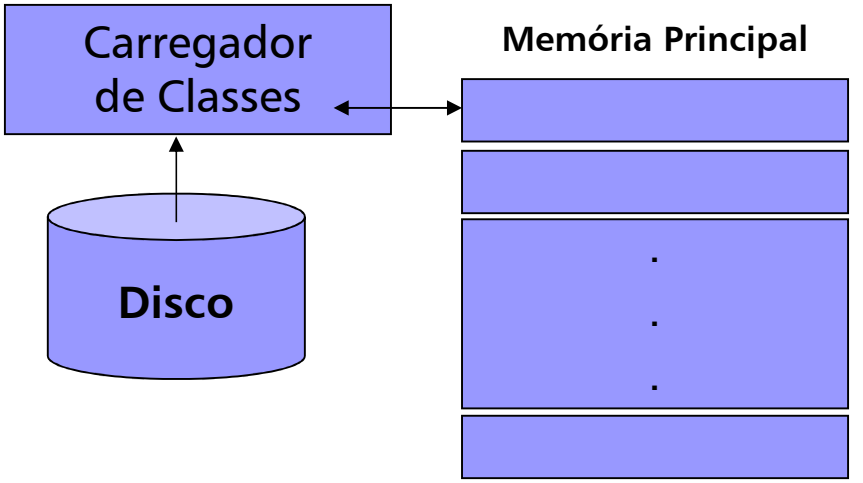
O programa é criado no editor e armazenado em disco.

Fase 2



O compilador cria bytecodes e os armazena em disco.

Fase 3

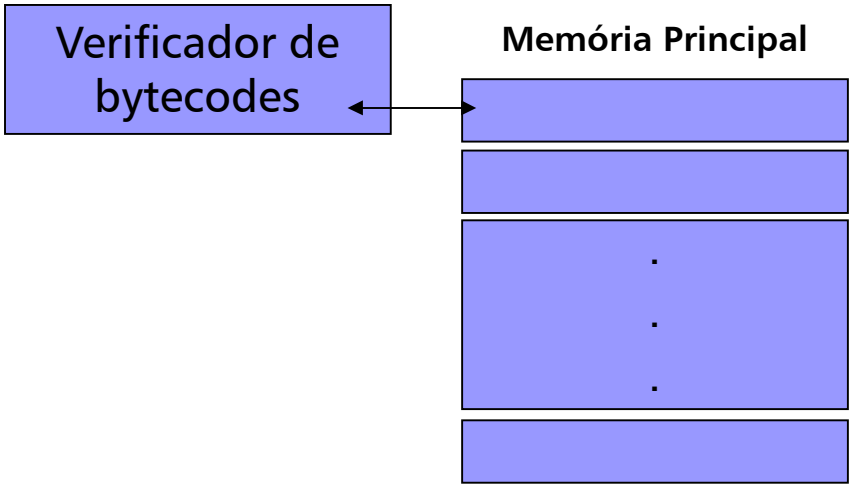


O carregador de classe coloca bytecodes na memória.

Fonte: Deitel & Deitel, 2003.

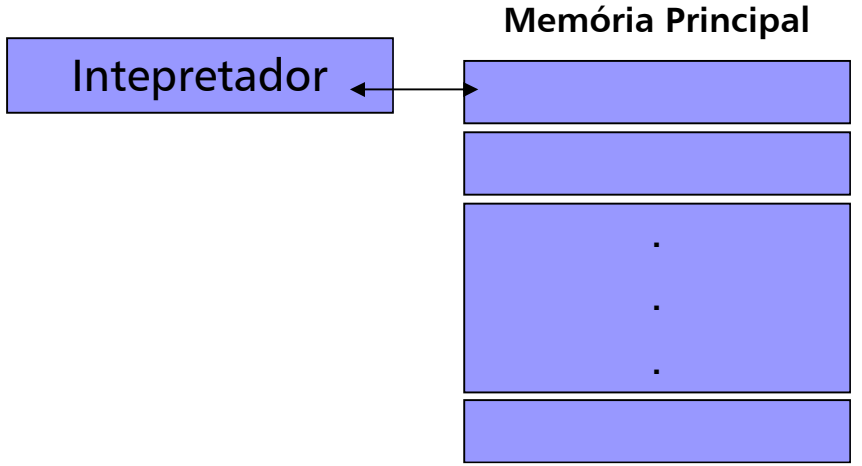
Visão geral

Fase 4



O verificador de bytecodes confirma que todos os bytecodes são válidos e não violam restrições de segurança do Java

Fase 5



O interpretador lê os bytecodes e os traduz para uma linguagem que o computador pode entender, possivelmente armazenando valores dos dados enquanto executa o programa.

Fonte: Deitel & Deitel, 2003.

Não se esqueça de preparar o ambiente

- Instale o SDK compatível com seu SO:
  - Download em http://java.sun.com.
- Crie uma variável de ambiente chamada JAVA_HOME, que deve guardar o caminho da pasta onde o Java foi instalado e adicione o caminho para os programas do Java no PATH do seu SO;
- No Windows:
  - SET JAVA_HOME=C:\JDK
  - SET PATH=%PATH%;%JAVA_HOME%\bin
- No Linux:
  - export JAVA_HOME=/usr/java/jdk
  - export PATH=$PATH:%JAVA_HOME%/bin