nemo

ontology & conceptual modeling research group

Desenvolvimento OO com Java

1 – Introdução

Vítor E. Silva Souza

(vitorsouza@inf.ufes.br)

http://www.inf.ufes.br/~vitorsouza

Departamento de Informática

Centro Tecnológico

Universidade Federal do Espírito Santo

Conteúdo

- Introdução;
- Tipos, variáveis e operadores;
- Estruturas de controle;
- Programação básica;
- Classes e objetos;
- Reuso de classes;
- Polimorfismo;
- RTTI e Interfaces;
- Classes Internas;
- Exceções;
- Arquivos e Streams;
- Utilitários;
- Tipos Genéricos.

O conteúdo dos slides é baseado na apostila de Java de autoria do prof. Flávio M. Varejão (fvarejao@inf.ufes.br), com colaboração do prof. Berilhes B. Garcia e de Rodrigo M. Pessoa (capítulos 1 a 6). Um link para download encontra-se disponível no site do curso.

A quem se destina este curso?

Pré-requisitos

- Saber o básico sobre lógica de programação;
- Conhecer alguma linguagem de programação.

Recomendações

- Perguntar SEMPRE que houver dúvida;
- Estudar a apostila em casa;
- Fazer todos os exercícios;
- Participar de um JUG.

Não é necessário nenhum conhecimento prévio da linguagem Java.

Objetivos do curso

- Tornar os alunos programadores capazes de:
  - Entender o paradigma orientado a objetos e construir soluções neste paradigma;
  - Utilizar Java para criar programas orientados a objeto, utilizando conceitos básicos e avançados da plataforma;
  - Ler e entender programas em Java escritos por outros programadores.

Por que Orientação a Objetos?

- Padrão para desenvolvimento de sistemas;
- Vantagens sobre paradigma estruturado:
  - Abstração mais próxima do mundo real;
  - Foco na reusabilidade;
  - Maior manutenibilidade;
  - Maior grau de qualidade da solução final.

Por que Java?

- Uma das linguagens OO mais usadas;
- Características de Java:
  - Simples, porém versátil, robusta e segura;
  - Portável (independente de sistema operacional);
  - Gratuita e open source;
  - Dirigida por uma especificação aberta;
  - Popular, rodeada por uma comunidade muito ativa;
  - De alta aceitação e com suporte da indústria;
  - Muitas ferramentas disponíveis;
  - Muita documentação disponível.

O que é Java?

- Uma tecnologia;
- Uma linguagem de programação;
- Uma plataforma de desenvolvimento;
- Um software distribuído pela Oracle;
- Um ambiente de execução de programas;
- Uma ilha da Indonésia (e o mar ao norte da ilha).

Java como linguagem de programação

- É o principal foco deste curso;
- Para entendermos e avaliarmos melhor Java como linguagem, veremos alguns conceitos da área de LP:
  - Propriedades desejáveis de LPs;
  - Tradução de programas;
  - Alocação de memória;
  - Abstração.

Propriedades desejáveis em LPs

- Foco no programador e não no programa;
- Legibilidade;
- Redigibilidade;
- Confiabilidade;
- Eficiência;
- Facilidade de aprendizado;
- Reusabilidade de código;
- Flexibilidade;
- Harmonia com a metodologia de projeto.

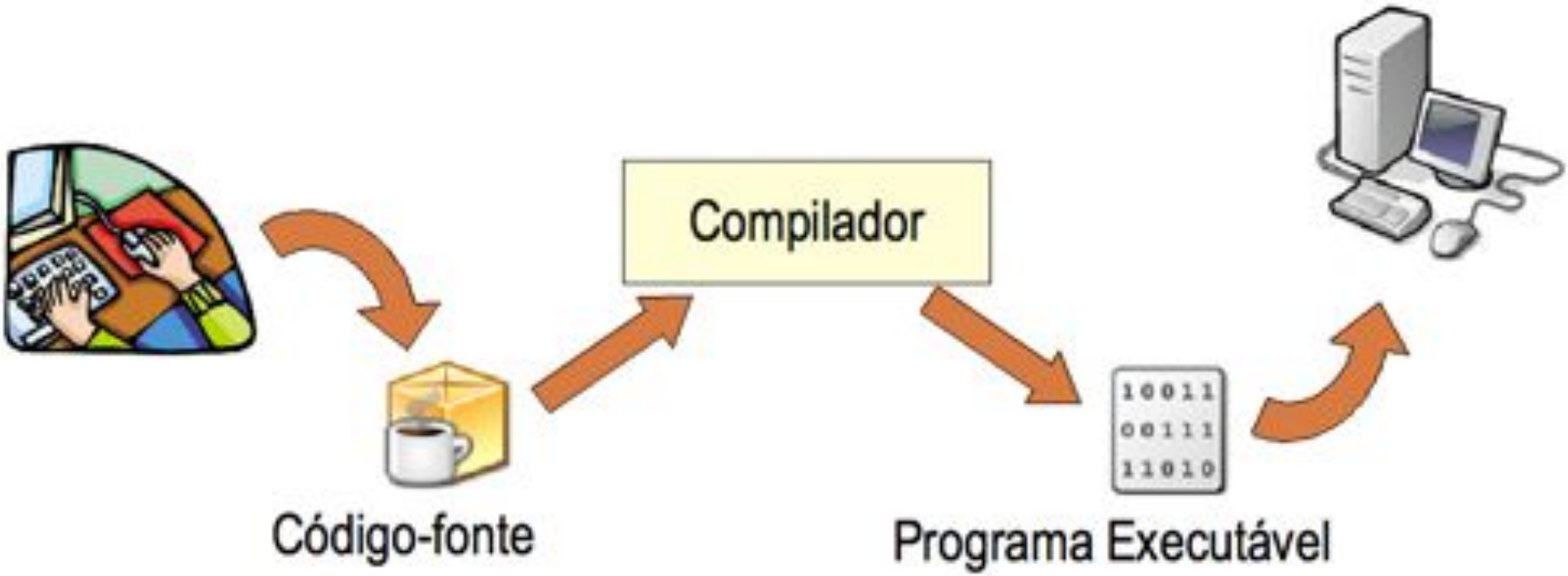
Funcionamento das linguagens

- O computador é um hardware que só entende operações muito básicas (zeros e uns);
- Programa executável = coleção de instruções em linguagem de máquina;
- Criar programas em linguagem de máquina é extremamente difícil e improdutivo;
- Usamos linguagens de programação de alto nível;
- Precisamos de um programa que transforme uma linguagem em outra: um tradutor.

Tradução de programas

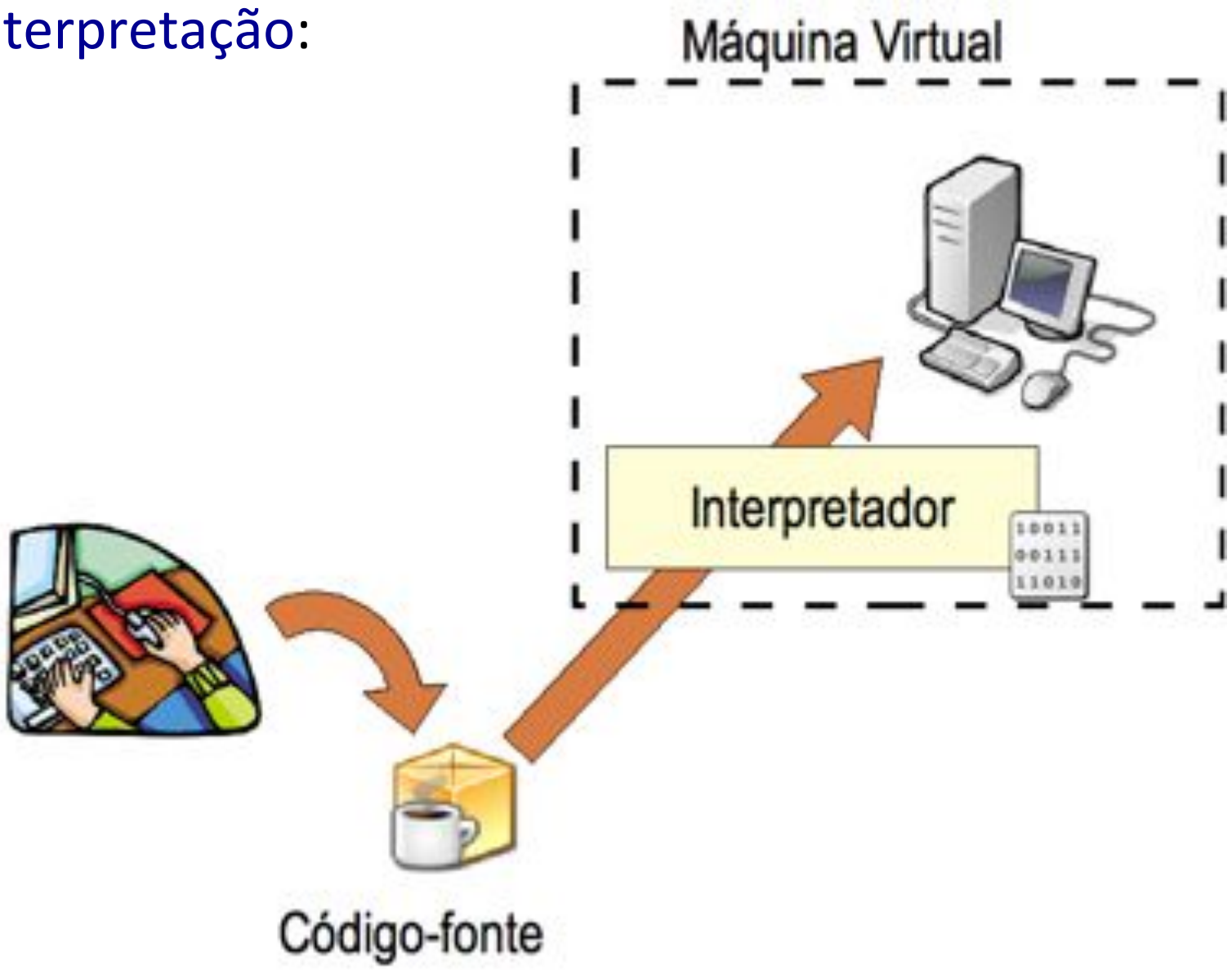
- Existe duas maneiras de se traduzir um programa: compilação e interpretação.

- Compilação:



Tradução de programas

- Interpretação:



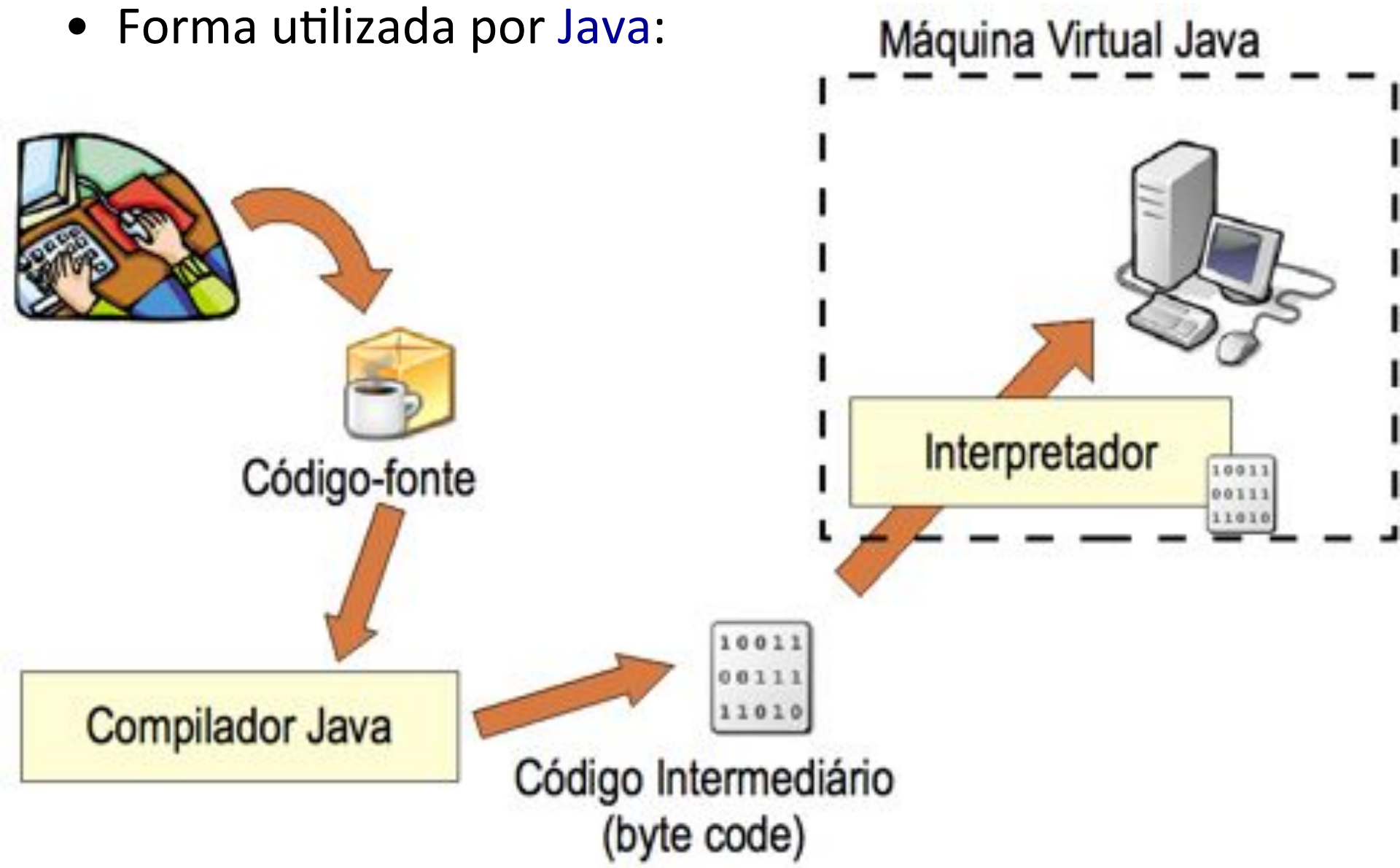
Compilação x interpretação

- Compilação:
  - Execução mais rápida;
  - Somente o executável é carregado em memória.
- Interpretação:
  - Portabilidade.

- Compilação + Interpretação = Híbrido
  - Une as vantagens (e desvantagens) de ambos.

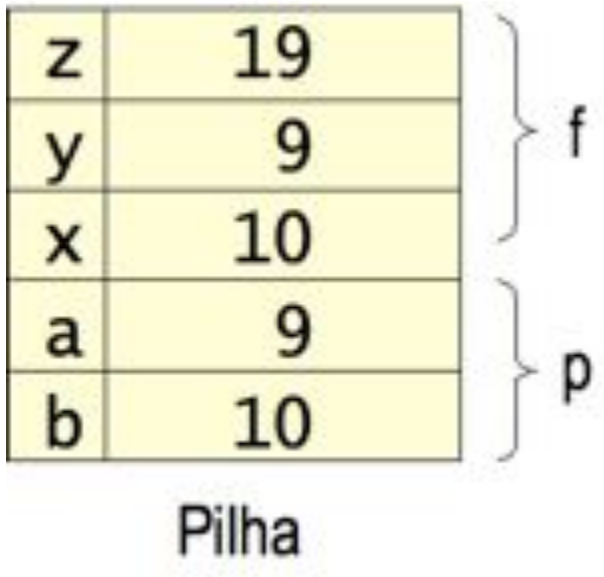
Tradução híbrida

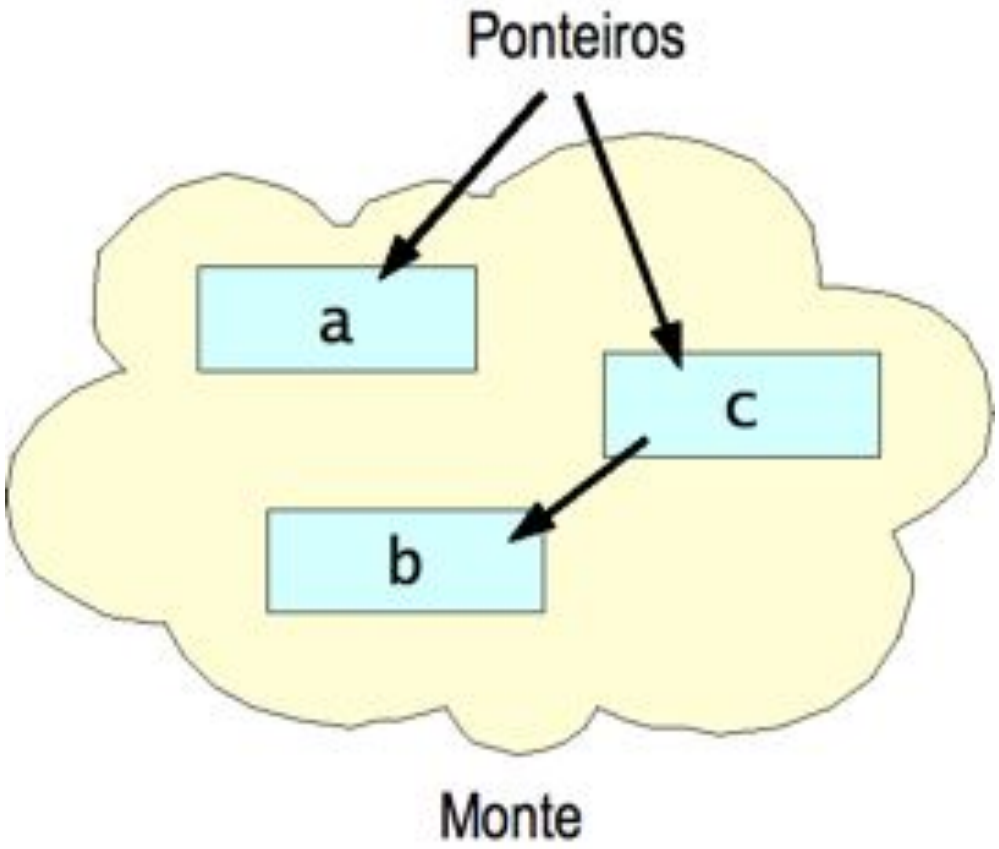
- Forma utilizada por Java:



Alocação de memória

- Estratégias de alocação:
  - Alocação estática;
  - Alocação dinâmica.



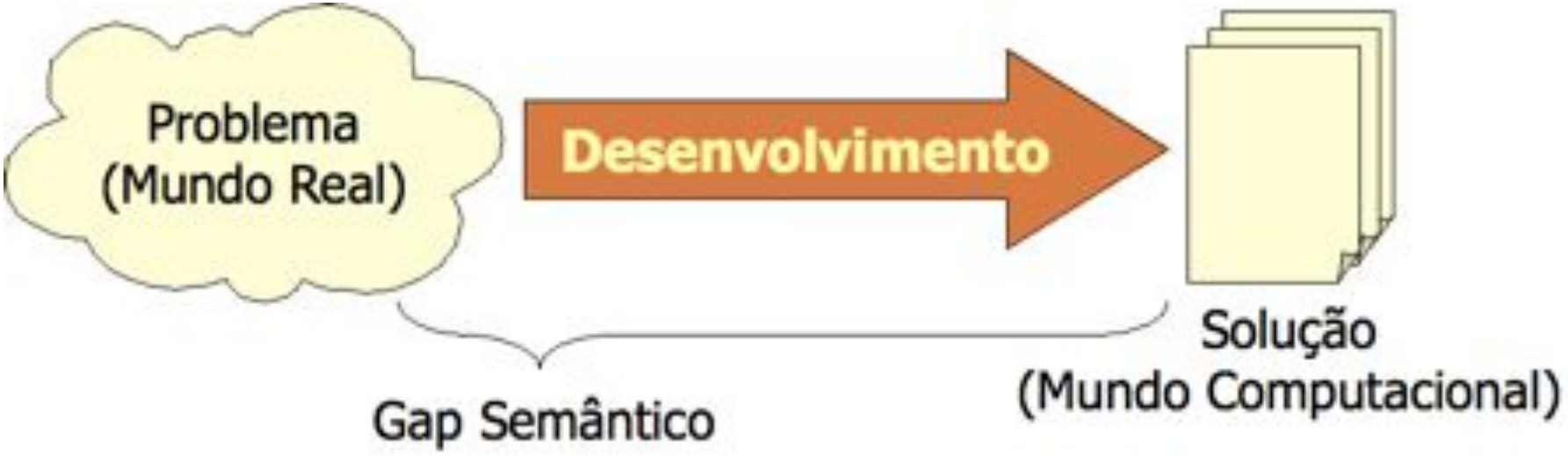


Decisões sobre alocação de memória

- Qual estratégia utilizar?
  - Por que usar pilha e monte?
  - Modelo de memória ALGOL-like.
- Quem é responsável pela alocação?
  - O programador (mais controle);
  - A própria linguagem (mais facilidade).
- O que o programador pode fazer com ponteiros?
  - Aritmética de ponteiros;
  - Ponteiros somente como referências.

Abstração

- Conceito fundamental em LPs:
  - Linguagem de máquina abstrai o hardware;
  - Linguagem de alto nível abstrai a de máquina;
  - Etc.
- O objetivo é escrever as soluções em termos cada vez mais próximos do mundo real.



Abstração e Orientação a Objetos

- Dentre os paradigmas existentes, a Orientação a Objetos destaca-se pelo nível de abstração:
  - Elementos do mundo real são modelados como objetos no mundo computacional;
  - Objetos possuem propriedades e comportamento, assim como no mundo real;
  - O código expressa a solução em termos mais próximos do problema.

Para um estudo mais aprofundado...

- Linguagens de Programação – Conceitos e Técnicas
  - Editora Elsevier (Campus);
  - Coleção Campus – SBC;
  - Flávio Miguel Varejão;
  - 2004.

História de Java

- 1995: Patrick Naughton e Sun Microsystems;
- Projeto Green – busca por inovação: *7;
- James Gosling e a linguagem Oak;
- Projeto Green muda de rumo depois que *7 não deslancha: a aposta é na Internet;
- Oak vira Java, graças a um café;
- Nascem as Applets, Java é incluída no Netscape, disponibilizada ao público e deslancha;
- Java fez 10 anos em 2005 e se tornou open source em 2006;
- Google introduz o sistema operacional Android em 2008;
- Oracle compra a Sun Microsystems em 2010;
- Java 8 lançado em 2014.

http://oracle.com.edgesuite.net/timeline/java/

A plataforma Java

- Plataforma = SO + Hardware:
  - Windows + PC (Intel / AMD);
  - Linux + PC;
  - MacOS X + Macintosh.
- Situa-se um nível acima do SO, formando uma nova plataforma de computação:
  - Portável (“WORA” - Write Once Run Anywhere);
  - Baseada na Máquina Virtual Java (JVM);
  - Linguagem Java é a parte central da plataforma;
  - 29 outras linguagens suportadas, incluindo Clojure, Groovy, Scala, JRuby,Jython, Rhino.

Características de Java

- Orientada a objetos:
  - Quase pura, pois possui tipos primitivos.
- Baseada em C++:
  - Sintaxe semelhante;
  - Porém mais simples.
- Portável:
  - Compilação para bytecode e interpretação na JVM;
  - Especificação rígida (JCP).

Características de Java

- Confiável:
  - Verificações na compilação e execução;
  - Incentiva-nos a escrever códigos melhores;
  - Não há aritmética de ponteiros, que são tratados como referências a objetos;
  - A gerência de memória é feita pela JVM (coletor de lixo), facilitando a tarefa do programador.
- Dinâmica:
  - Classes são carregadas sob demanda (class loader).

Características de Java

- Projetada para ambientes distribuídos:
  - Suporte de alto nível para construção de aplicações em rede (sockets, RMI, etc.);
  - Com carregamento dinâmico, classes podem ser obtidas da rede e acionadas em tempo de execução;
  - "The network is the computer".
- Segura:
  - Verificações em tempo de execução;
  - Verificação de bytecode;
  - Modelo sandbox (caixa de areia);
  - Assinatura digital e criptografia.

Características de Java

- Possui bom desempenho:
  - Linguagens híbridas não têm o mesmo desempenho de linguagens compiladas;
  - No entanto, existem diversas otimizações (ex.: JIT), com melhorias a cada nova versão;
  - Em última instância, integra-se com códigos em C.
- Facilita a programação concorrente:
  - Dispõe de elementos que facilitam a programação de sistemas com uso intensivo de threads paralelas.

Edições de Java

- Java é distribuída em três edições:
  - Java Standard Edition (Java SE);
  - Java Enterprise Edition (Java EE);
  - Java Mobile Edition (Java ME).

Java SE

- Ferramentas de desenvolvimento e API núcleo da plataforma (base para as demais);
- Permite o desenvolvimento de aplicações desktop, com interface gráfica, acesso à bancos de dados, I/O, acesso à rede, etc.;
- Dividida em:
  - JDK = Java Development Kit;
  - JRE = Java Runtime Environment.

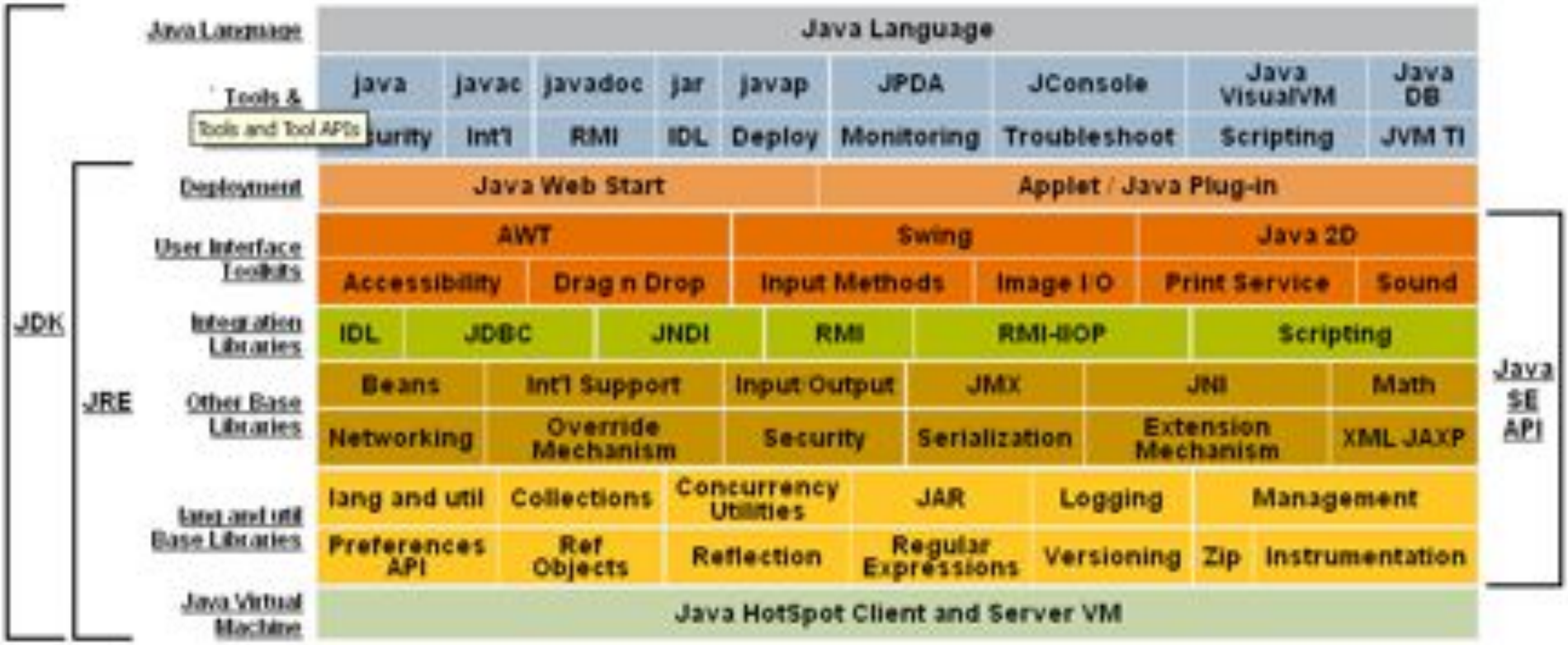
Java Development Kit

- Somente para programadores;
- Contém:
  - Ferramentas de desenvolvimento;
  - Ambiente de execução (JRE);
  - API Java SE (compilada e código-fonte);
  - Programas de demonstração;
  - Bibliotecas adicionais;
  - Documentação (obtida separadamente).

Java Runtime Environment

- Necessária para rodar programas Java (bytecodes compilados);
- É a única parte da plataforma Java que os clientes precisam instalar;
- Em alguns SOs pode vir instalada;
- A Oracle provê suporte oficial às plataformas Windows, Mac OS, Solaris e Linux.

A API Java SE


Java Language
Java Language
java
javac
javadoc
jar
javap
JPDA
JConsole
Java VisualVM
Java DB
Tools & Tools and Tool APIs
Security
Int'l
RMI
IDL
Deploy
Monitoring
Troubleshoot
Scripting
JVM TI
Deployment
Java Web Start
Applet / Java Plug-in
User Interface Toolkits
AWT
Swing
Java 2D
Accessibility
Drag n Drop
Input Methods
Image I/O
Print Service
Sound
Integration Libraries
IDL
JDBC
JNDI
RMI
RMI-IIOP
Scripting
Other Base Libraries
Beans
Int'l Support
Input Output
JMX
JNI
Math
Networking
Override Mechanism
Security
Serialization
Extension Mechanism
XML JAXP
lang and util Base Libraries
lang and util
Collections
Concurrency Utilities
JAR
Logging
Management
Preferences API
Ref Objects
Reflection
Regular Expressions
Versioning
Zip
Instrumentation
Java Virtual Machine
Java HotSpot Client and Server VM
JDK
JRE
Java SE API

A API Java SE

- AWT/Swing: interfaces gráficas;
- Java2D: desenho;
- JDBC: acesso a bancos de dados;
- JNDI: acesso a servidores de nomes;
- RMI: invocação remota de métodos ;
- i18n: suporte à internacionalização;
- I/O: entrada e saída (arquivos);
- JNI: integração com linguagens nativas;
- Math: cálculos matemáticos;

A API Java SE

- Networking: transmissão de dados via rede;
- Security: segurança;
- Serialization: persistência por serialização;
- XML: processamento de XML e afins;
- Lang & Util: núcleo da linguagem, utilitários;
- Concurrency: programação concorrente;
- Logging: funções de relatório (log);
- Reflection: RTTI (reflexão, introspecção).

Java EE

- Permite o desenvolvimento de aplicações corporativas:
  - Multi-camadas, distribuídas, centradas em servidores, altamente robustas, estáveis e escaláveis.
- Inclui as especificações para desenvolvimento Web: Servlets, JSP, Web Services, JSF, etc.;
- Inclui especificações da plataforma Enterprise Java Beans (EJB):
  - Componentes gerenciados integrados a outras tecnologias Java EE para prover acesso remoto, persistência e transações transparentes, etc.

Java EE

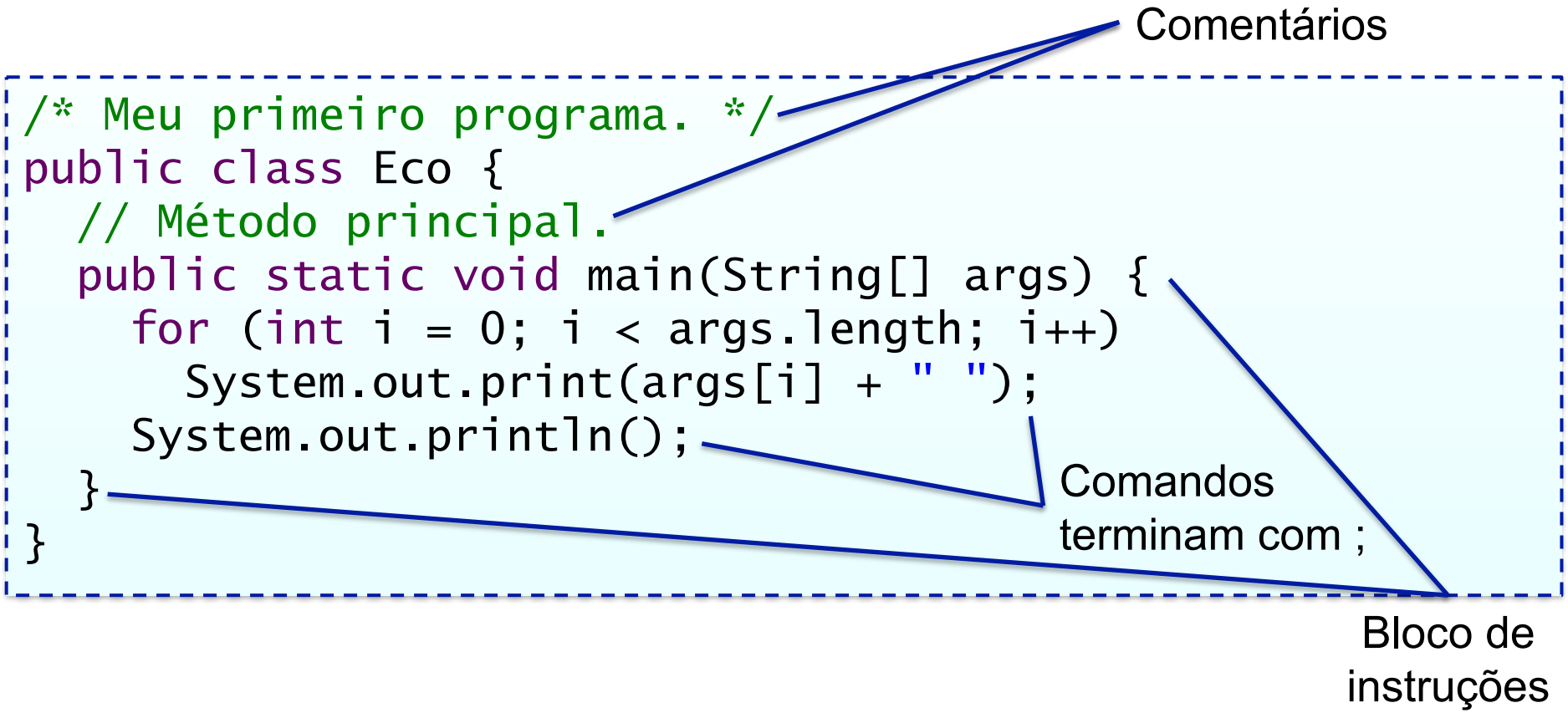
- Outras tecnologias relacionadas:
  - RMI/IIOP e Java IDL: conectividade;
  - JDNI: servidor de nomes;
  - JAC e JNI: acesso a sistemas legados;
  - JAAS: segurança;
  - JTA: transações em bancos de dados;
  - JMS e JavaMail: envio de mensagens;
  - E tudo o mais que temos no Java SE...

Java ME

- Permite o desenvolvimento de aplicações para dispositivos móveis:
  - Telefones celulares;
  - PDAs (Palm, iPaq, etc.);
  - Dispositivos embarcados (embedded);
  - Etc.
- Em grande parte vem sendo substituída pela plataforma Android;
- Java Card: aplicações para Smart Cards e outros dispositivos muito limitados.

Um primeiro programa

- Escreva o seguinte programa:



- Salve como Eco.java (case sensitive).

Compilando o programa

Código-fonte: Eco.java

```
public class Eco {
  // ...
}
```

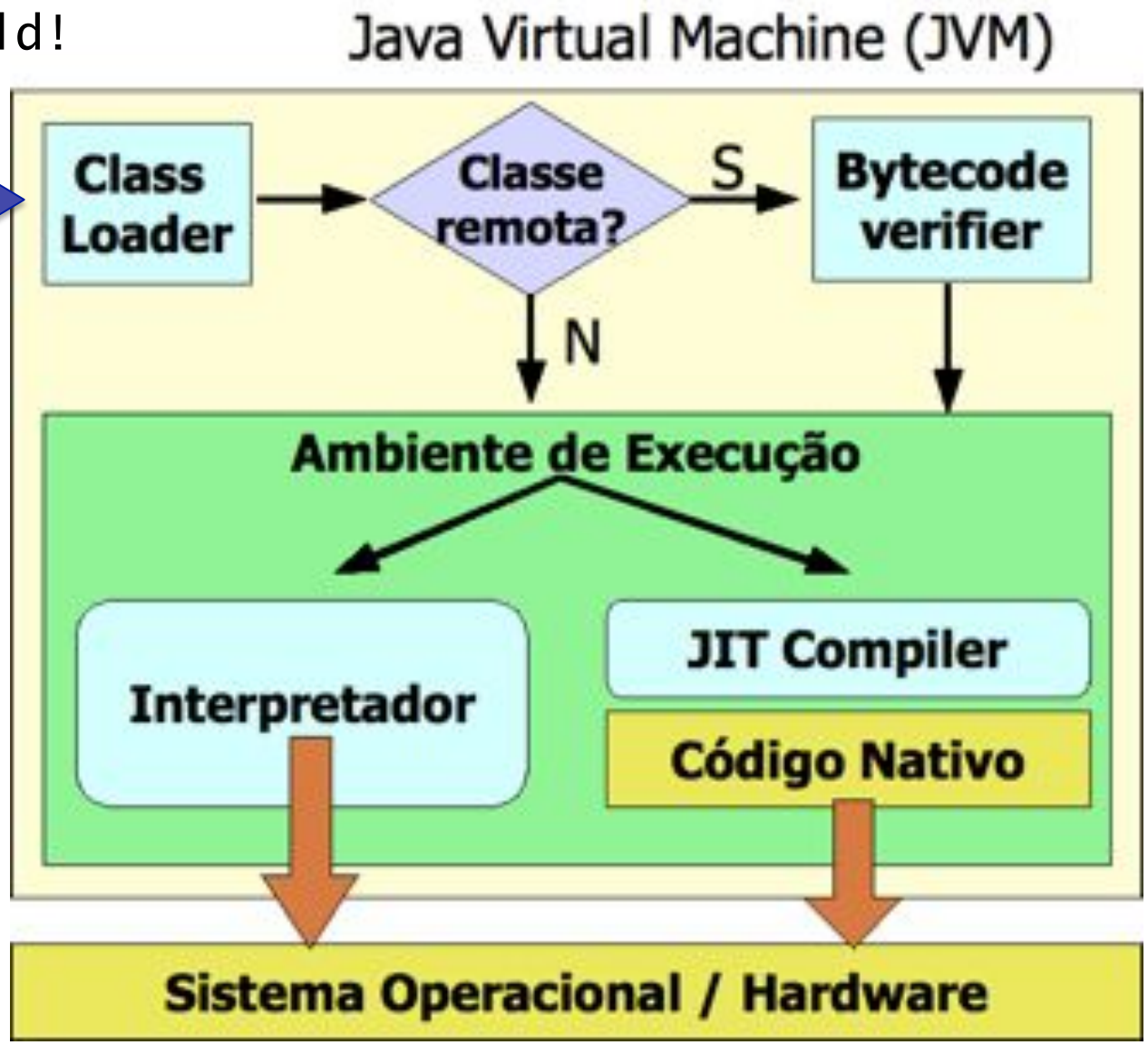
javac Eco.java

Bytecode: Eco.class

```
CA FE BA BE 00 00 00 33 00 2C 0A 00 0B
00 15 09 00 16 00 17 07 00 18 0A 00 03
00 15 0A 00 03 00 19 08 00 1A 0A 00 03
00 1B 0A 00 1C 00 1D 0A 00 1C 00 1E ...
```

Executando o programa

```
java Eco Hello, World!
```




```
CA FE BA BE 00
00 00 33 00 2C
0A 00 0B 00 15
09 00 16 00 ...
```

Bytecode: Eco.class

Programas Java

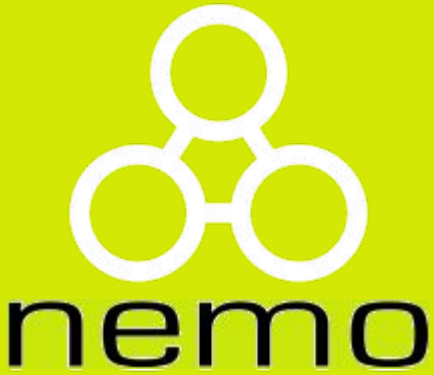
- Um programa Java é uma classe pública com o método `main()`, como no nosso exemplo;
- O nome do arquivo deve coincidir com o nome da classe que possui o método `main()`;
- Pode haver mais de uma classe no mesmo arquivo fonte, mas somente uma pode ser pública;
- Veremos estes conceitos ao longo do curso.

IDEs

- Ambientes integrados de desenvolvimento facilitam o trabalho de programação:
  - Eclipse (http://www.eclipse.org);
  - NetBeans (http://www.netbeans.org);
  - IntelliJ IDEA (http://www.jetbrains.com/idea);
  - JDeveloper (http://www.oracle.com/technetwork/developer-tools/jdev/);
  - Dentre outras...

Recursos para aprender Java

- Comunidades virtuais:
  - http://www.portaljava.com
  - http://www.guj.com.br
  - http://www.javafree.com.br
- JUGs – Grupos de Usuários Java
  - http://www.esjug.org
- Revistas:
  - Mundo Java;
  - Java Magazine.
- Livros.



http://nemo.inf.ufes.br/