



UNIVERSIDADE FEDERAL
DO ESPÍRITO SANTO

Centro Tecnológico
Departamento de Informática

Prof. Vítor E. Silva Souza

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

[Desenvolvimento OO com Java]

O que é Java?



Esta obra está licenciada com uma licença Creative Commons Atribuição-
Compartilhado 4.0 Internacional: <http://creativecommons.org/licenses/by-sa/4.0/>.

Conteúdo do curso



O que é Java;

- Variáveis primitivas e controle de fluxo;
- Orientação a objetos básica;
- Um pouco de vetores;
- Modificadores de acesso e atributos de classe;
- Herança, reescrita e polimorfismo;
- Classes abstratas e interfaces;
- Exceções e controle de erros;
- Organizando suas classes;
- Utilitários da API Java.

Estes slides foram baseados na [apostila do curso FJ-11: Java e Orientação a Objetos da Caelum](#) e na apostila Programação Orientada a Objetos em Java do [prof. Flávio Miguel Varejão](#).

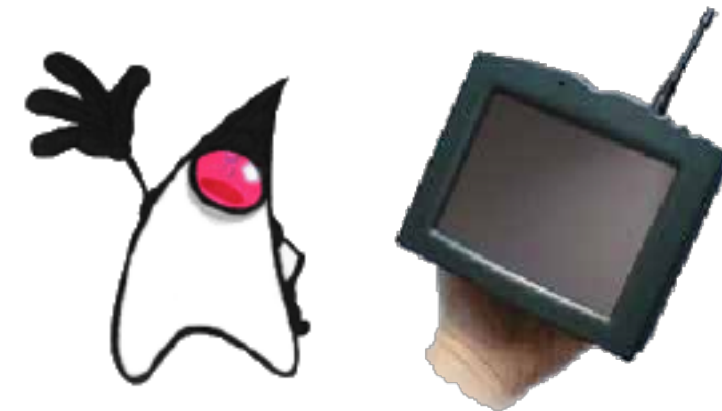
O que é Java?

- 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).



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 deslança: [Internet](#);
- Oak vira [Java](#), graças a um café;
- Nascem as [Applets](#), Java é incluída no Netscape, disponibilizada ao público e deslança;
- 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 2009/2010;
- [Java 8](#) lançado em 2014, [Java 9](#) em 2017.

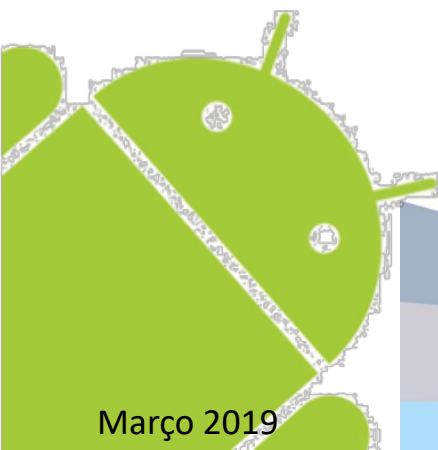


História de Java

- Mudanças de mercado:
 - *Java foi idealizada para o mercado de TVs a cabo e outros aparelhos eletrodomésticos;*
 - *Java foi lançada com foco nos clientes web (Applets);*
 - *Hoje Java tem destaque do lado do servidor e em aparelhos celulares.*



<http://oracle.com.edgesuite.net/timeline/java/>
<http://www.java.com/en/javahistory/>

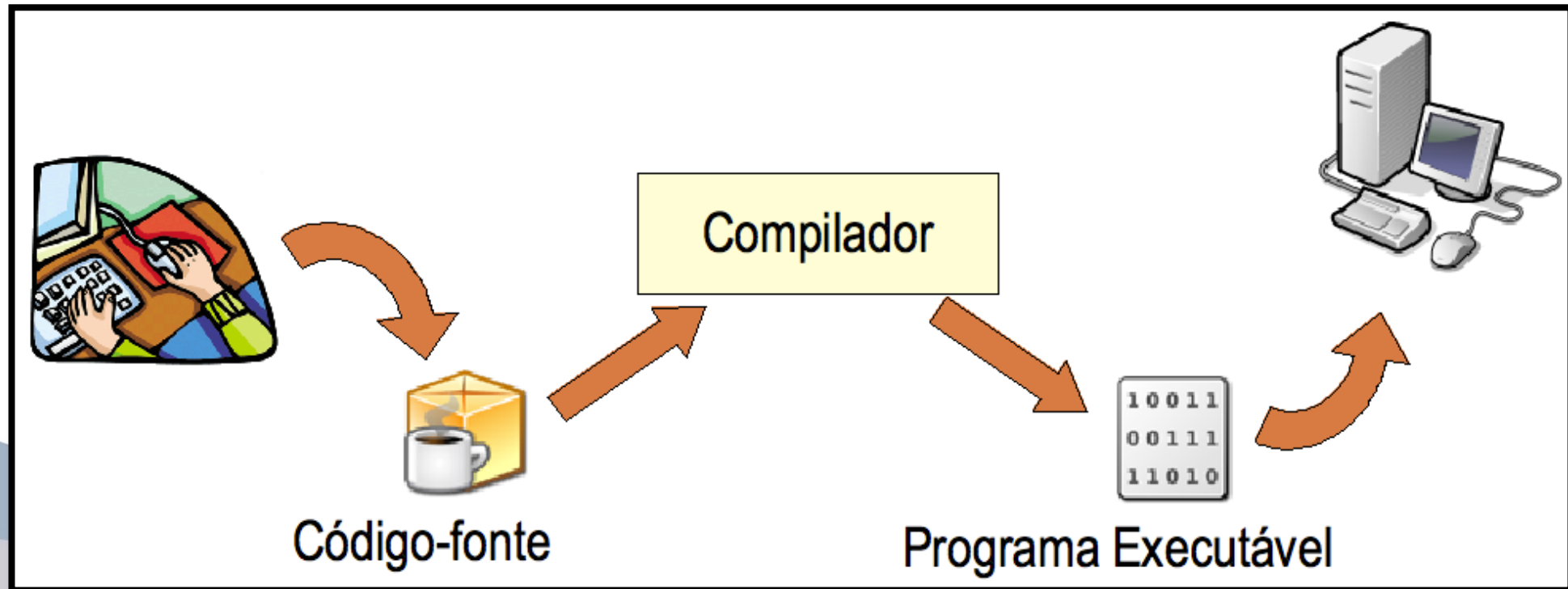


Contexto em que Java surge

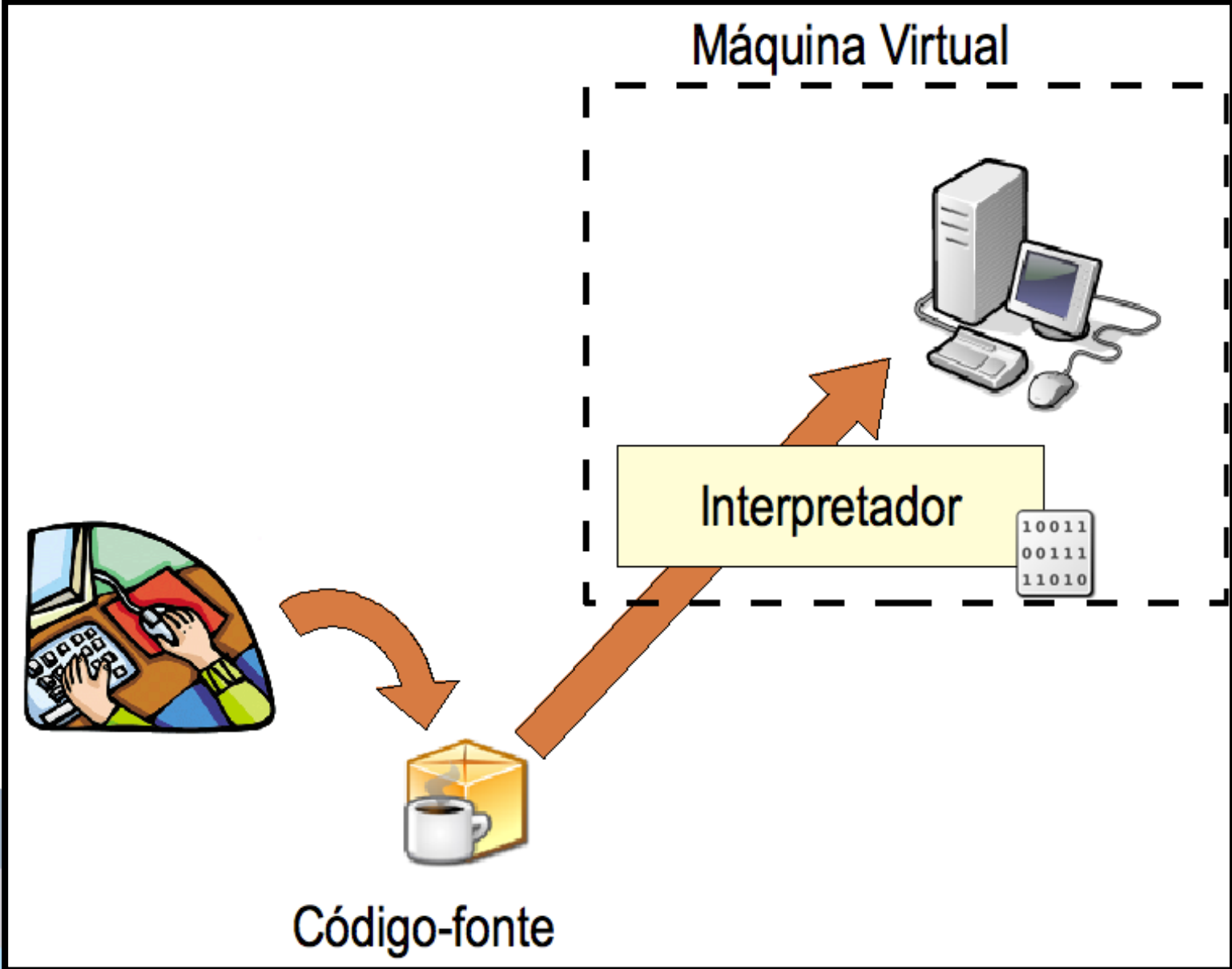
- Você está cansado de...
 - *ter que manipular **ponteiros**?*
 - *ter que **alocar/desalocar** memória?*
 - *ter que **organizar** arquivos em diretórios e controlar seus Makefiles?*
 - *ter que escrever **utilitários** para coisas muito básicas?*
 - *ter que **reescrever** parte do código ao mudar de **SO**?*
 - *ter que **pagar** para usar a tecnologia de desenvolvimento?*

Máquina Virtual Java

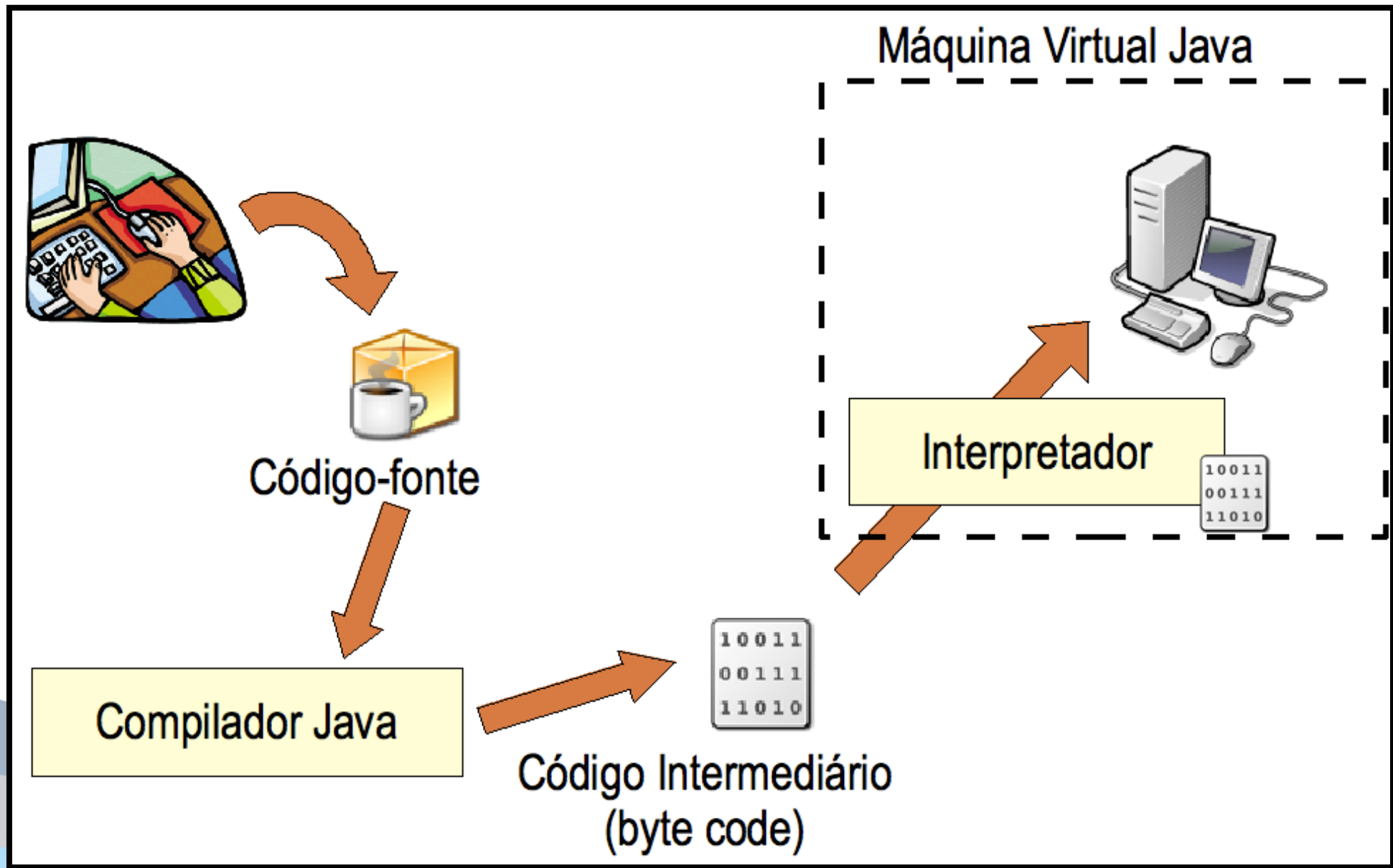
- Existem **duas maneiras** de se traduzir um programa: compilação e interpretação.
- Compilação:**



Máquina Virtual Java



Máquina Virtual Java



Vantagens da JVM

- Portabilidade;
- Isola a aplicação do SO;
- Pode tirar métricas e realizar otimizações (HotSpot e JIT Compiler);
- Se ocorre um erro, fecha-se a máquina virtual, sem afetar outras JVMs ou o SO;
- 29 **outras linguagens** suportadas, incluindo Clojure, Groovy, Scala, JRuby, Jython, Rhino, etc.



O código de máquina gerado por um compilador Java é conhecido por “bytecode”, pois existem menos de 256 códigos de operação dessa linguagem e cada “opcode” gasta um byte. (Apostila FJ-11 Caelum)

Versões e edições de Java

- Standard Editions:
 - *Java 1.0 (1996);*
 - *Java 1.1 (1997);*
 - *J2SE 1.2 (1998);*
 - *J2SE 1.3 (2000);*
 - *J2SE 1.4 (2002);*
 - *Java 1.5 / Java 5 (2004);*
 - *Java SE 6 (2006);*
 - *Java SE 7 (2011);*
 - *Java SE 8 (2014);*
 - *Java SE 9 (2017);*
 - *Java SE 10 (2018);*
 - *Java SE 11 (2018, LTS);*
 - *Java SE 12 (2019).*
- Enterprise Editions:
 - JPE project (1998);
 - J2EE 1.2 (1999);
 - J2EE 1.3 (2001);
 - J2EE 1.4 (2003);
 - Java EE 5 (2006);
 - Java EE 6 (2009);
 - Java EE 7 (2013);
 - Java EE 8 (2017).

Cedido pela Oracle para a fundação
Eclipse em 2018, agora Jakarta EE.

Novas versões todo ano em março e
setembro (*long term support*).

• Mobile Editions...

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:
 - *JRE = Java Runtime Environment;*
 - *JDK = Java Development Kit.*

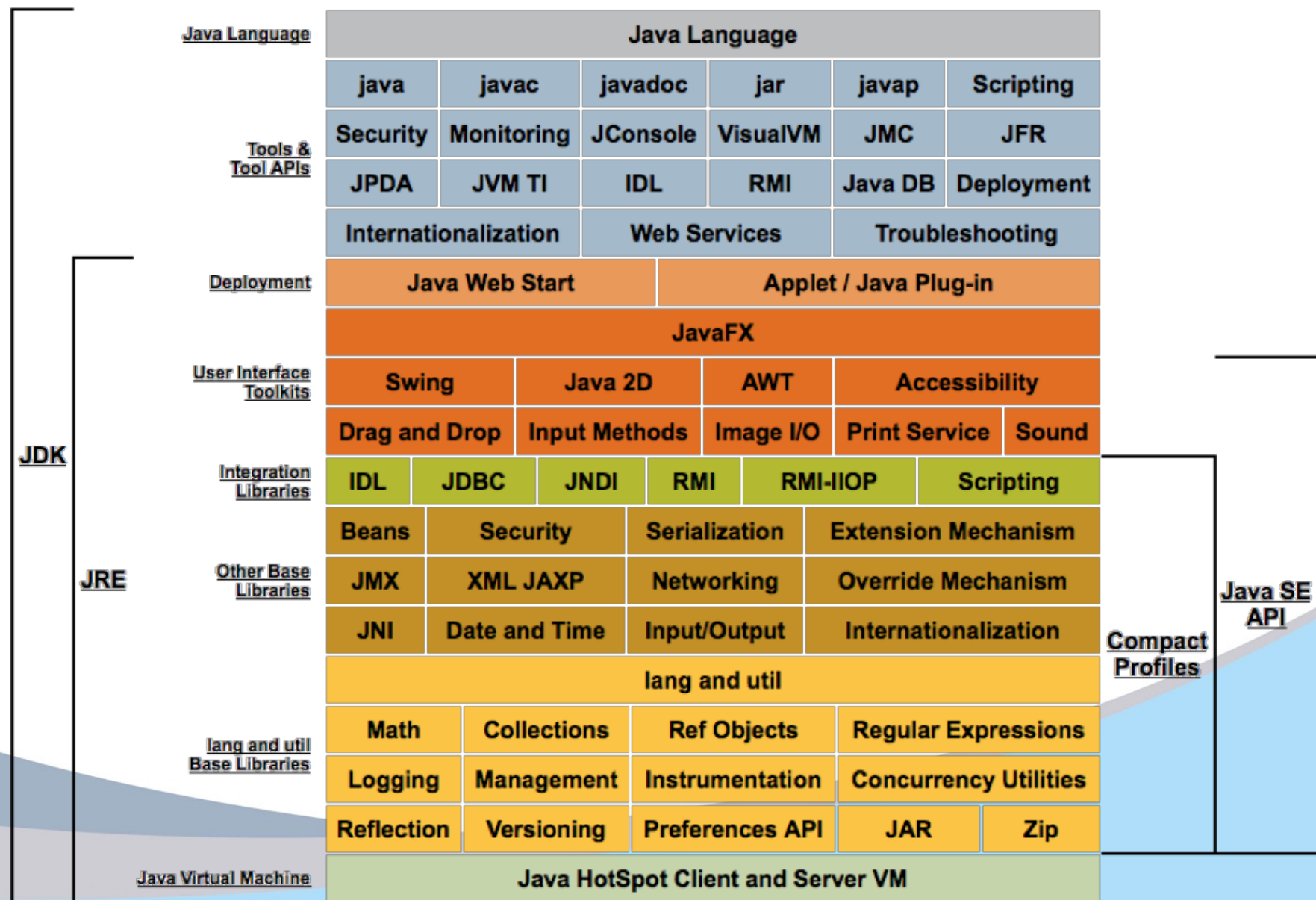
Java Runtime Environment

- JVM + bibliotecas básicas da API;
- 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.

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)*.

A API Java SE



Java EE

- Permite o desenvolvimento de aplicações corporativas:
 - *Multicamadas, 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 ME

- Permite o desenvolvimento de aplicações para dispositivos **móveis**:
 - *Telefones celulares;*
 - *PDA's (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.

Por que Java?

- Foco em aplicações de **médio/grande porte**;
- Enorme **ecossistema**: muitas bibliotecas disponíveis;
- Alta **legibilidade**, conectividade, **portabilidade**;
- Grande oferta de **empregos**;
- Gratuita e **open source**;
- Independência de fabricante:
especificação aberta;
- **Popular**, rodeada por uma **comunidade** muito ativa;
- De alta **aceitação** e com suporte da **indústria**;
- Muitas ferramentas e documentação disponíveis.



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

- **Dinâmica:**
 - *Classes são carregadas sob **demanda** (class loader);*
- **Confiável:**
 - ***Verificações** na compilação e execução;*
 - *Incentiva-nos a escrever **códigos melhores**;*
 - *Não há aritmética de **ponteiros**;*
 - *Gerência de **memória** feita pela JVM (**coletor de lixo**);*

Características de Java

- Segura:
 - *Verificações de **bytecode**, modelo **sandbox**;*
 - *Assinatura digital e **criptografia**;*
- Facilita a programação **concorrente**:
 - *Dispõe de elementos que **facilitam** a programação de sistemas com uso intensivo de **threads paralelas**;*

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”;*

Características de Java

- Possui bom **desempenho**:
 - *Não era o caso das primeiras versões...*
 - **Otimizações** (ex.: JIT), com **melhorias** a cada nova versão (melhor que C/C++ em alguns benchmarks);
 - Em última instância, **integra-se** com códigos em **C**.

Um primeiro programa

- Escreva o seguinte programa:

```

/* Meu primeiro programa. */
public class Eco {
    // Método principal.
    public static void main(String[] args) {
        for (int i = 0; i < args.length; i++)
            System.out.print(args[i] + " ");
        System.out.println();
    }
}

```

Comentários

Comandos terminam com ;

Bloco de instruções

- Salve como Eco.java (*case sensitive*).

Programas Java

- Java é uma linguagem um pouco **burocrática**:
 - 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;
- Calma! Veremos estes conceitos ao longo do curso...

Compilando o programa

Código-fonte: Eco.java

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

javac Eco.java



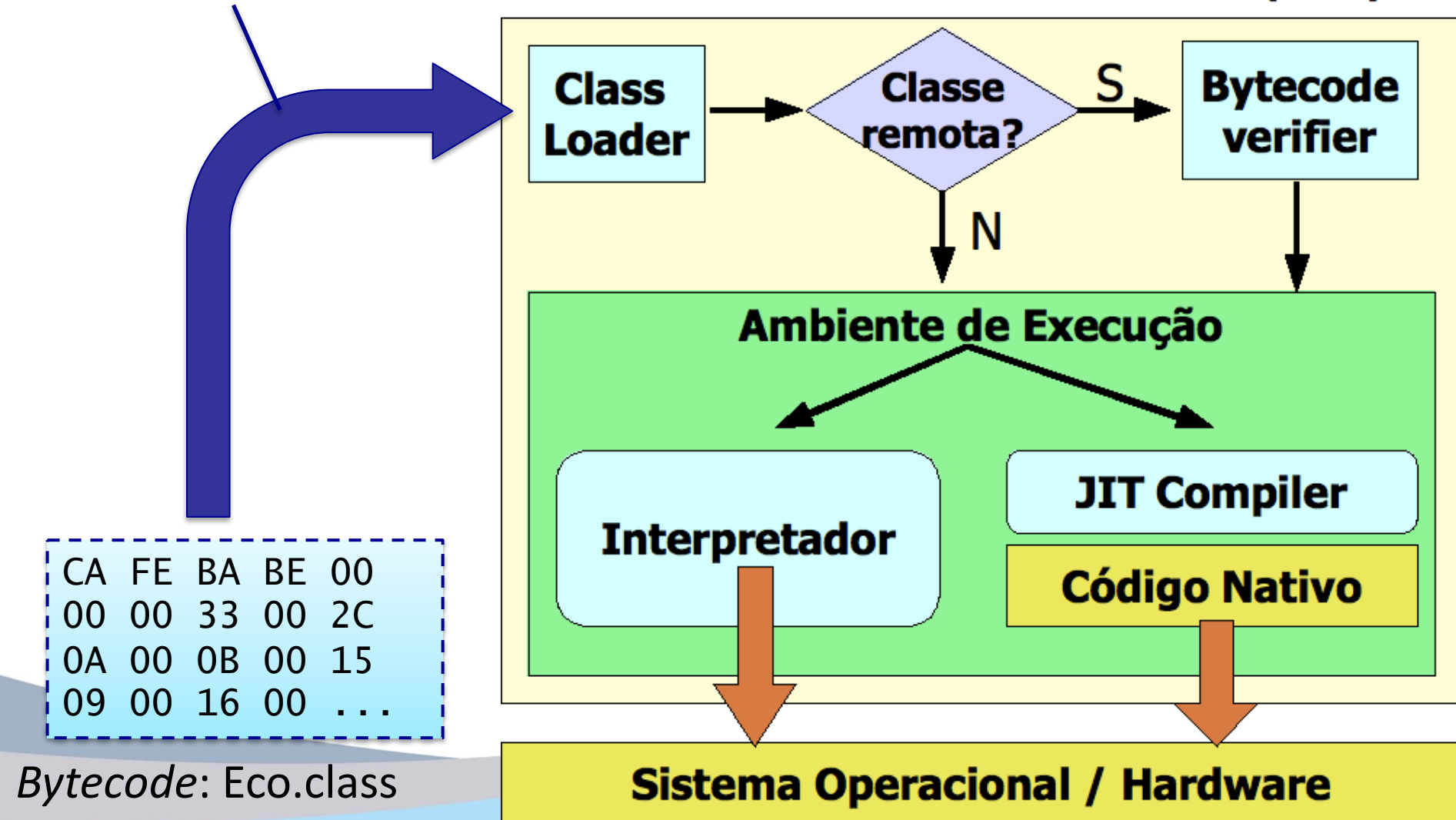
Bytecode: Eco.class

```
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!

Java Virtual Machine (JVM)



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 e fóruns:
 - *<http://www.guj.com.br>*
 - *<http://stackoverflow.com>*
- **JUGs** – Grupos de Usuários Java:
 - ~~*<http://www.esjug.org>*~~
- **Revistas**:
 - *Java Magazine ([.com.br](http://www.oracle.com/javamagazine));*
 - *Java Magazine ([Oracle](http://www.oracle.com/javamagazine)).*
- **Apostilas e livros.**