Universidade Estadual de Maringá
Departamento de Informática

Tecnologia Java

Conteúdo baseado nos materiais dos Professores:
Marcos Aurélio Domingues (DIN/UEM)
Edson Oliveira Junior (DIN/UEM)
Michelle Nery (IF Sul de Minas)
Bruno Boniati (UFMS)

Prof.ª Juliana Keiko Yamaguchi
março de 2019

Objetivos

- O que é Java?
- História e evolução.
- Prós e contras.
- Funcionamento.
- Tecnologias relacionadas.
- Ambientes de desenvolvimento integrado.

Java

- O que é Java?

Write once, run anywhere!

Escreva uma vez, execute em qualquer lugar!

Java

- Java é uma linguagem de programação orientada a objetos desenvolvida pela Sun Microsystems.

Java

- Modelada depois de C++, a linguagem Java foi projetada para ser pequena, simples e portável a todas as plataformas e sistemas operacionais, tanto o código fonte como os binários.
- Java é multiplataforma. Isto quer dizer que não é necessário usar um tipo especifico de computador, não importa se você usa Windows, Mac ou Unix.

Java
Breve histórico

- Em 1991, a Sun Microsystems, de olho no crescente mercado de dispositivos eletrônicos (eletrodomésticos) inteligentes voltados ao consumidor, financiou um projeto interno (*Green Project*).

Java
Breve histórico

- A ideia era produzir uma linguagem de computador reduzida e simples, e que gerasse um código eficiente para ser utilizado em dispositivos com algumas restrições:
  - Pouca memória;
  - Diferentes CPUs;
  - O software produzido não poderia se limitar a uma única arquitetura.

Java
Breve histórico

- James Gosling (chefe da equipe) batizou a linguagem de Oak.
  - Oak significa carvalho, em inglês, uma árvore que ele observava muito de sua janela.
  - Foi criado um mascote para o projeto, o Duke.

https://en.wikipedia.org/wiki/James_Gosling#/media/File:James_Gosling_2008.jpg

Java
Breve histórico

- O nome Oak já era patenteado então a linguagem foi batizada de Java em homenagem a Ilha de Java que produzia o café que a equipe da Sun consumia.

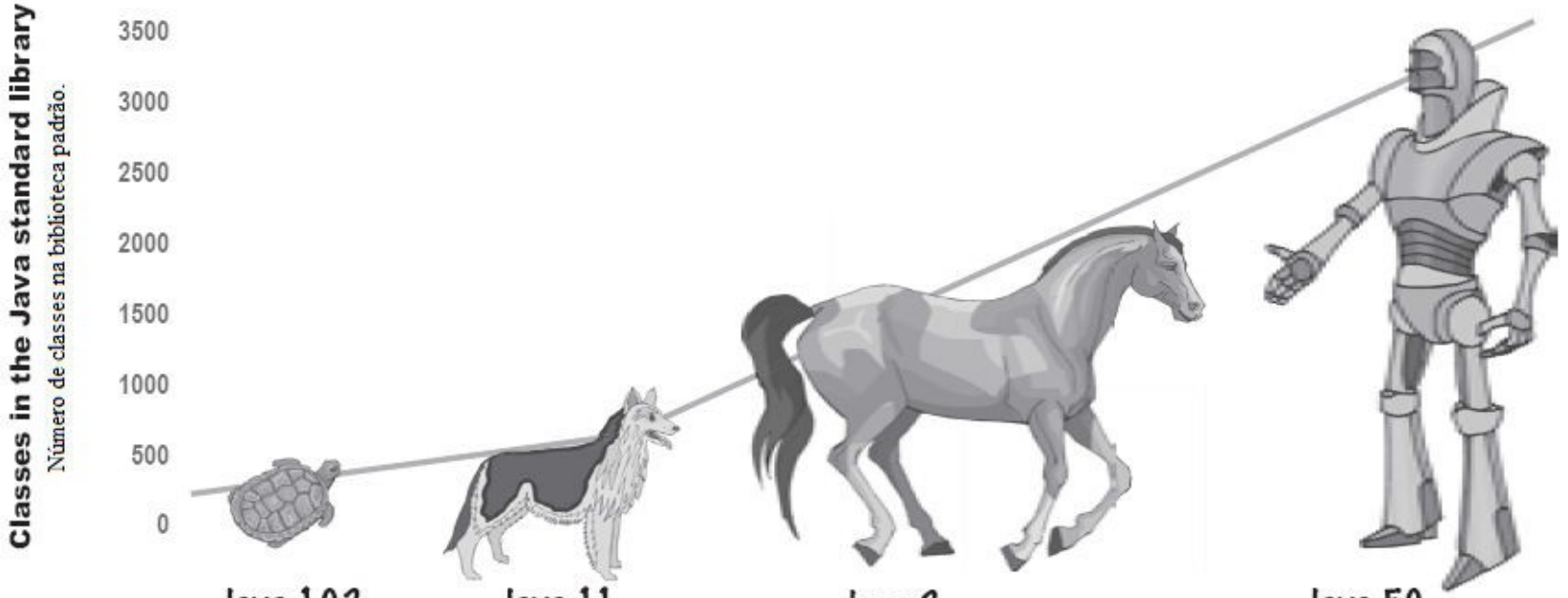
Java
Breve histórico

- Na época, o mercado de dispositivos eletrônicos interativos não teve um crescimento interessante como o esperado.
- No entanto, ao mesmo tempo ocorre uma explosão de popularidade da World Wide Web e este parece ser um mercado potencial para a linguagem Java.

Java
Breve histórico

- Em 1995 Java é anunciado formalmente.
- Em 2009 a Oracle adquire a SUN por US$ 7,4 bilhões e é quem mantém o Java atualmente.

Java
Evolução



Java 1.0.2

250 classes

Slow.
(Lento)

Nome e logotipo bonitos. Divertido de usar. Muitos bugs. Os Applets eram a grande novidade.

Java 1.1

500 classes

A ***little*** faster.
(Um pouco mais rápido)

Maior capacidade, mais amigável. Tornou-se bastante popular. Código para interface gráfica melhorado.

Java 2 (versions 1.2 – 1.4)

2300 classes

Much faster.
(Bem mais rápido)

Pode, as vezes, rodar em velocidade nativa. Bem mais poderosa. Dá suporte às plataformas: J2SE, J2ME e J2EE. Tornou-se uma linguagem robusta para aplicações Web, corporativas e móveis.

Java 5.0 (versions 1.5 and up)

3500 classes

More power, easier to develop with.
(Mais poderosa, mais fácil para desenvolver aplicações)

Além de mais de 1000 novas classes, Java 5.0 (Tiger) adicionou novas mudanças à linguagem Java, tornando-a mais fácil de programar por meio de novas caracteristicas amplamente conhecidas em outras linguagens.

Java
Características – Prós

- Portabilidade: escreva uma vez, execute em qualquer lugar.
- Escrever somente para a plataforma JAVA:
  - Suporte nos browsers da Web, TVs e celulares.
- Confiança:
  - A inexistência de ponteiros evita o acesso direto a memória.
- Internacionalização:
  - Linguagem de programação que suporta Unicode.

Java
Características – Prós

- Programas dinâmicos e extensíveis:
  - Código Java organizado em unidades modulares orientadas por objeto chamadas classes;
  - Cada classe é armazenada em um arquivo separado e somente descarregadas para o interpretador Java quando necessárias;
  - O código torna-se uma coleção interativa de componentes independentes de software.

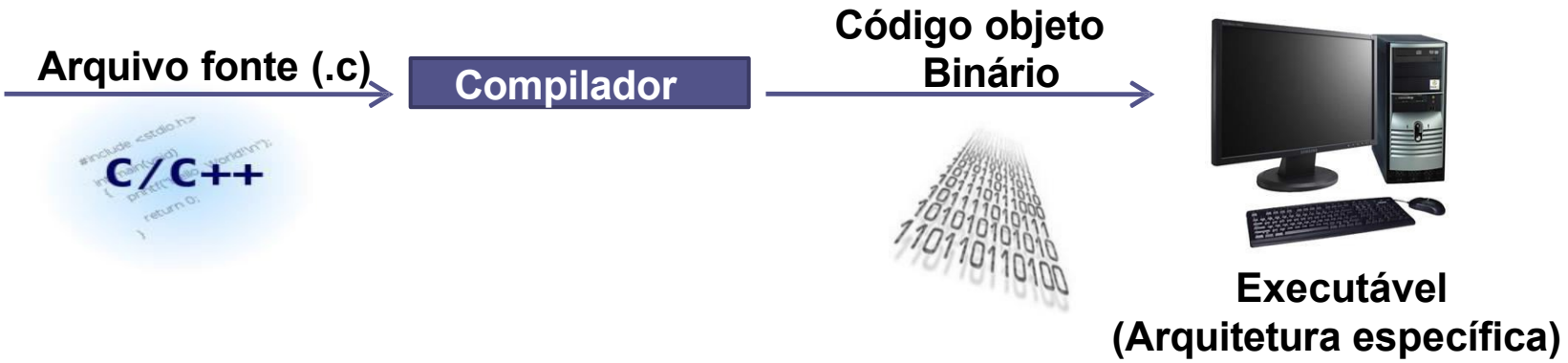
Java
Características – Prós

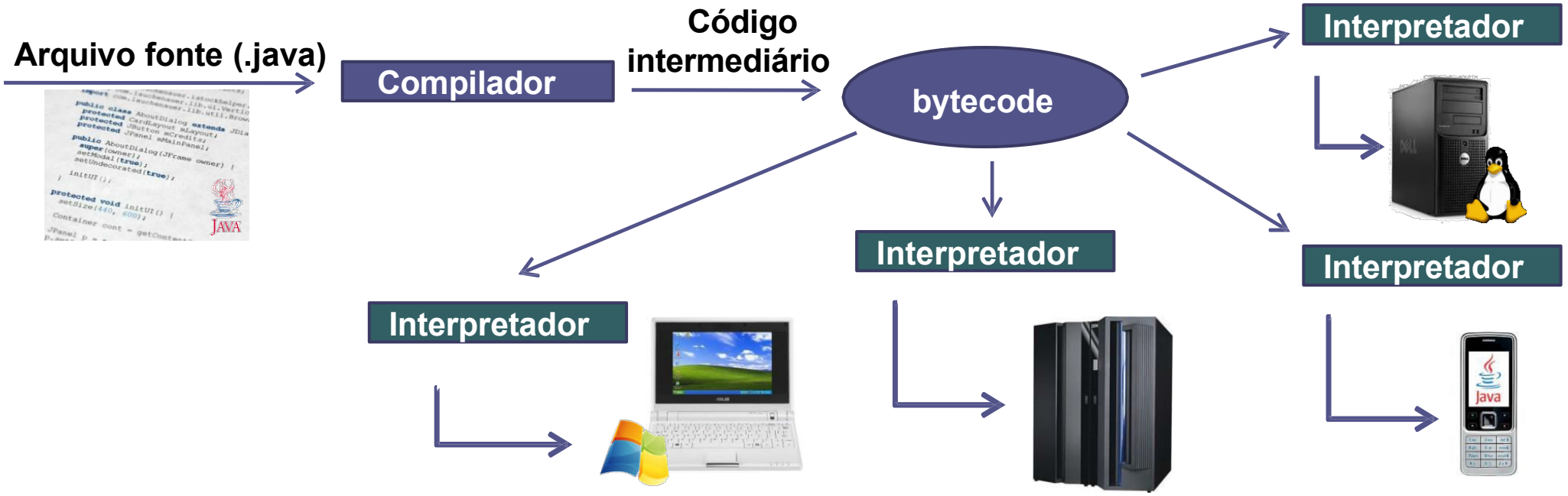
- Eficiência para o programador:
  - Conjunto de APIs (Application Programming Interface) poderoso e bem projetado.
- Torna as páginas da Web mais interessantes:
  - Som, vídeo, animações, relógios, contadores, etc.
- Multi plataforma:
  - Pode-se criar uma grande variedade de aplicações.

Java
Características – Contras

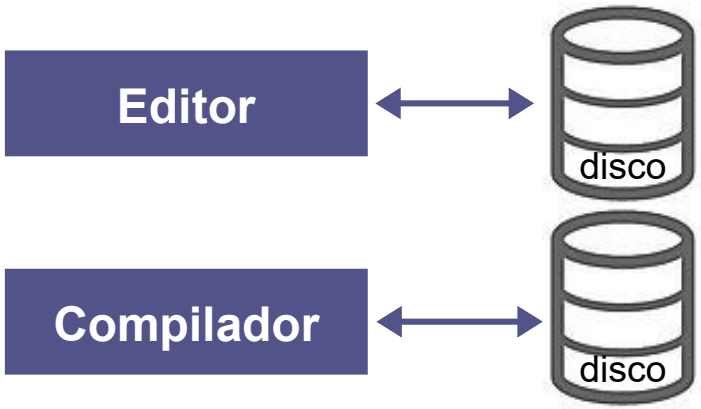
- Desempenho
  - O formato de implementação da linguagem Java é híbrida, ou seja, é um misto de compilada e interpretada.
  - O compilador gera um código chamado bytecode (arquivo .class) que é interpretado pela JVM (Java Virtual Machine).
  - Esse processo de pré-compilação pode afetar a sua eficiência (depende do poder da máquina).
  - Por ser interpretada, torna-se mais lenta, não podendo ser comparada à velocidade de execução de código nativo.

Java
Características



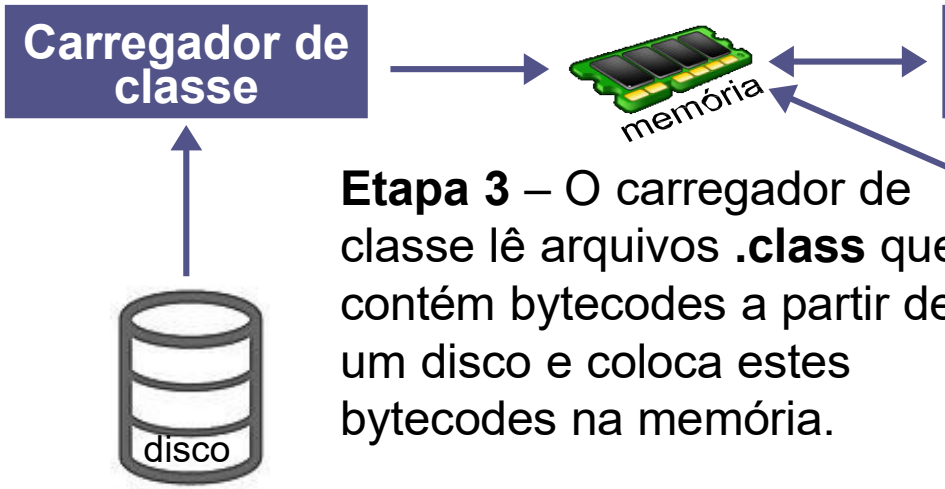


Java
Características



Etapa 1 – O programa é criado em um editor e armazenado no disco com a extensão **.java**.

Etapa 2 – O compilador cria bytecodes e os armazena em disco com a extensão .**class**.



Etapa 3 – O carregador de classe lê arquivos **.class** que contém bytecodes a partir de um disco e coloca estes bytecodes na memória.

Etapa 4 – O verificador de bytecodes confirma que todos os bytecodes são válidos e não violam restrições de segurança do Java.

Etapa 5 – Para executar o programa, a JVM lê os bytecodes e os traduz para uma linguagem de baixo nível específica para a plataforma onde o mesmo está sendo executado.

Máquina Virtual Java (JVM)

- Responsável por executar um programa Java, permitindo sua portabilidade.
- Processador virtual:
  - Possui seu próprio conjunto de instruções;
  - Manipula diferentes áreas de memória.
- Responsável por gerenciar:
  - memória (coletor de lixo), erros, exceções, threads.

Máquina Virtual Java (JVM)
Portabilidade

- A portabilidade é obtida pelo fato da linguagem ser interpretada, ou seja, o compilador gera um código independente de máquina chamado bytecode.
- No momento da execução, este bytecode é interpretado pela máquina virtual instalada na máquina real.

Máquina Virtual Java (JVM)
Portabilidade

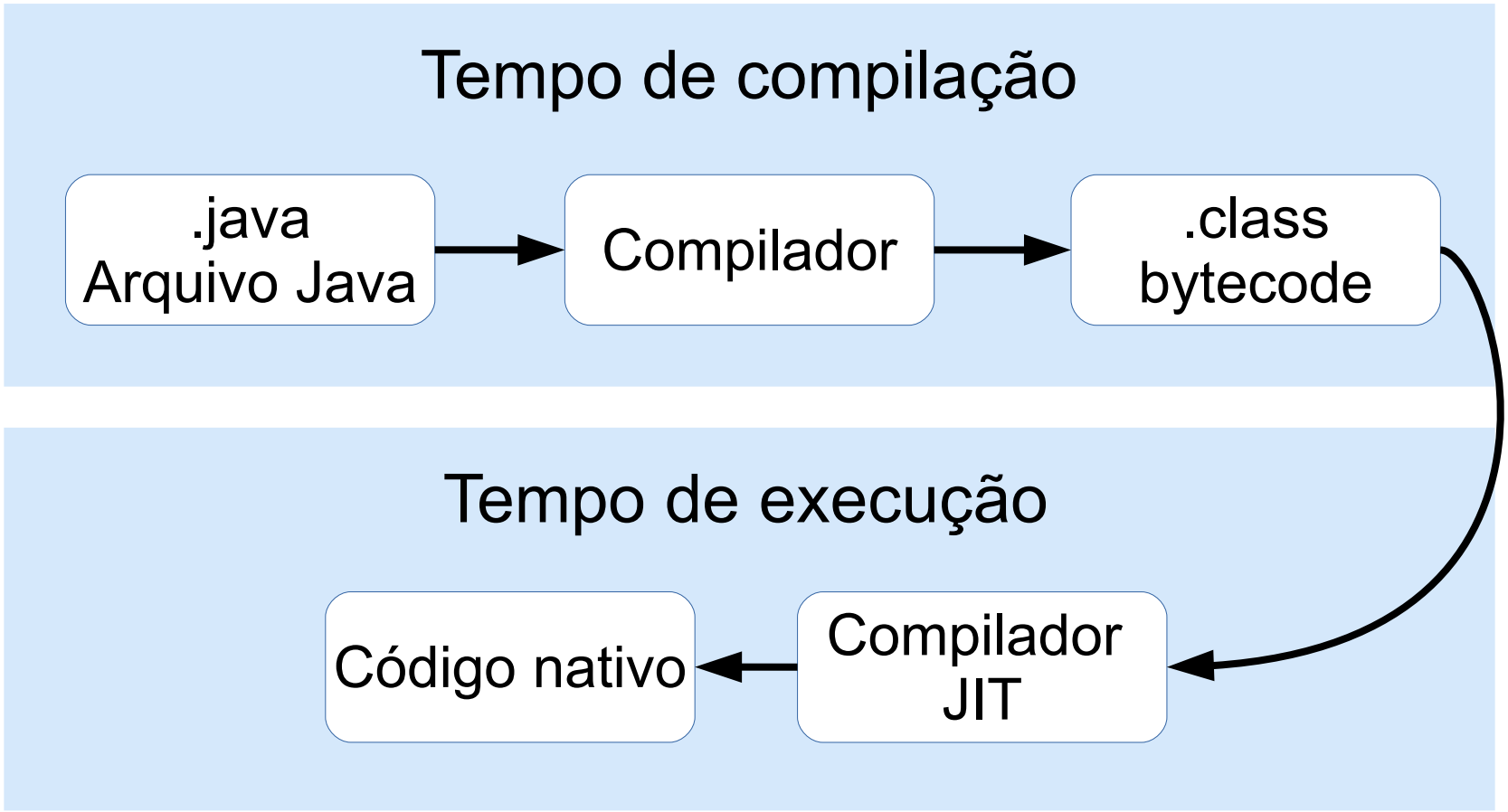
- Assim, o bytecode é um tipo de linguagem interpretada, que passa pelo processo de compilação e, em seguida, é interpretado por uma máquina virtual.
- Para portar Java para uma arquitetura hadware/SO específica, basta instalar a máquina virtual (interpretador).

Máquina Virtual Java (JVM)

- Garante para a linguagem Java:
  - Eficiência (Just In Time)
    - O bytecode é convertido para instruções nativas da máquina.
    - Just In Time significa compilar e traduzir trechos prestes a serem executados, otimizando o código dos mais frequentes.
  - Segurança
    - Antes de executar um código, é feito uma verificação no bytecode.
  - Portabilidade.

Máquina Virtual Java (JVM)

- Compilação Just In Time (JIT)



https://aboullaite.me/understanding-jit-compiler-just-in-time-compiler/

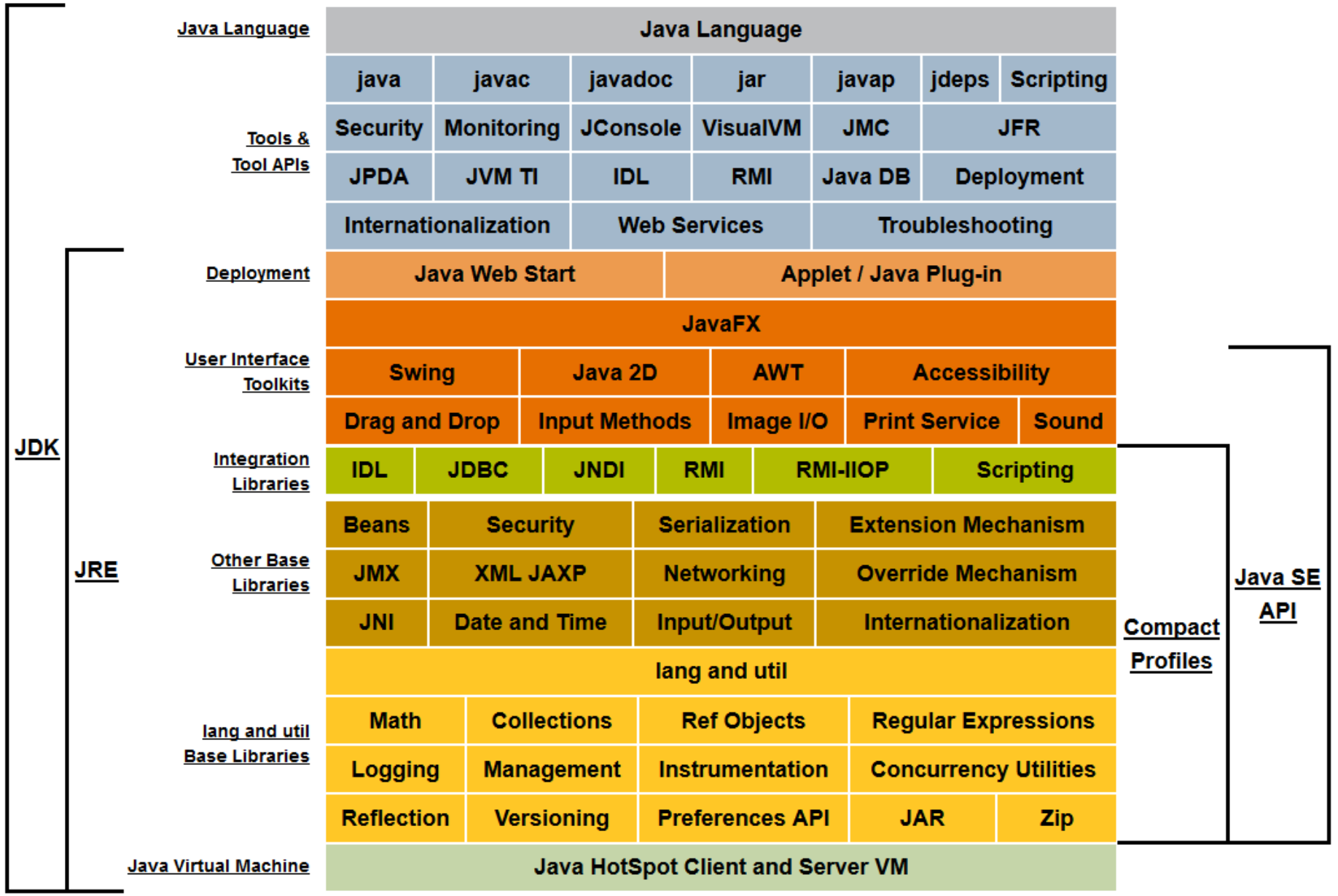
https://www.ibm.com/support/knowledgecenter/pt-br/SSYKE2_7.0.0/com.ibm.java.lnx.70.doc/diag/understanding/jit_overview.html

Máquina Virtual Java (JVM)

- A Máquina Virtual Java (JVM) não “conhece” a linguagem Java:
  - A JVM entende apenas arquivos em um formato binário particular (bytecode): arquivos “.class”.
- Permite rodar outras linguagens desde que seja possível traduzi-las para arquivos class:
  - Haskell, Pascal, Ada, etc.

Plataforma Java

versão 8 – http://docs.oracle.com/javase/8/docs/



APIs de Java
JRE e JDK

- JRE (*Java Runtime Environment*) é composto pelas API's necessárias para uma aplicação feita em Java (Platform SE) funcionar.
  - Plugins de navegadores,
  - Internet banking,
  - Aplicativos (exemplo: IRPF)
  - Jogos,
  - etc.

APIs de Java
JRE e JDK

- JDK (*Java Development Kit*) contém as APIs do JRE e as ferramentas necessárias para desenvolver aplicação utilizando a linguagem Java.
  - Compilador (javac)
  - Interpretador de bytecode (java)
  - Empacotador (jar)
  - Gerador de documentação (javadoc)
  - Debugger (jdb)

APIs de Java

- Uma API (*Application Programming Interface*) é uma coleção de classes (bibliotecas) contendo funcionalidades prontas para serem usadas no desenvolvimento de uma aplicação.
- Existem 3 tipos básicos de APIs para as seguintes plataformas:
  - Java SE (Standard Edition)
  - Java EE (Enterprise Edition)
  - Java ME (Micro Edition)

APIs de Java

- Plataformas Java
  - SE (Standard Edition): contém as APIs para o desenvolvimento e execução de aplicações Java.
  - EE (Enterprise Edition): inclui APIs para o desenvolvimento de aplicações para a Web
  - ME (Micro Edition) : voltada para desenvolvimento de sistemas embarcados
  - dispositivos de propósito específico: automóveis, celulares, eletrodomésticos, eletrônicos

Java SE

- Ela contém todo o ambiente necessário para a criação e execução de aplicações Java, incluindo a Máquina Virtual Java (JVM), o compilador Java, as APIs do Java e outras ferramentas utilitárias.
- Seu uso é voltado para computadores pessoais e servidores, onde há bem mais necessidade de aplicações.
- Mais indicada para quem quer aprender a linguagem.

Java EE

- O JEE é a plataforma Java voltada para redes.
- Contém bibliotecas especialmente desenvolvidas para o acesso a servidores, a sistemas de e-mail, a banco de dados, entre outras características.
- Desenvolvido para suportar uma grande quantidade de usuários simultâneos.

Java EE

- A plataforma JEE contém uma série de especificações:
  - JDBC (Java Database Connectivity), utilizado no acesso e conexão ao banco de dados;
  - JSP (Java Server Pages), uma espécie de página Web (aplicações para Internet);
  - Servlets que permite a geração de requisições e respostas.
  - Servidores Web que permitem a configuração de páginas na Web.

Java ME

- O JME (Java Micro Edition) é o ambiente de desenvolvimento para dispositivos móveis ou portáteis, como telefones celulares, TVs e palmtops.
- Desde que seus dispositivos tenham uma JVM (Java Virtual Machine - Máquina Virtual Java), é possível, com poucas modificações, implementar os aplicativos em qualquer aparelho, sendo o único limite a capacidade do hardware:

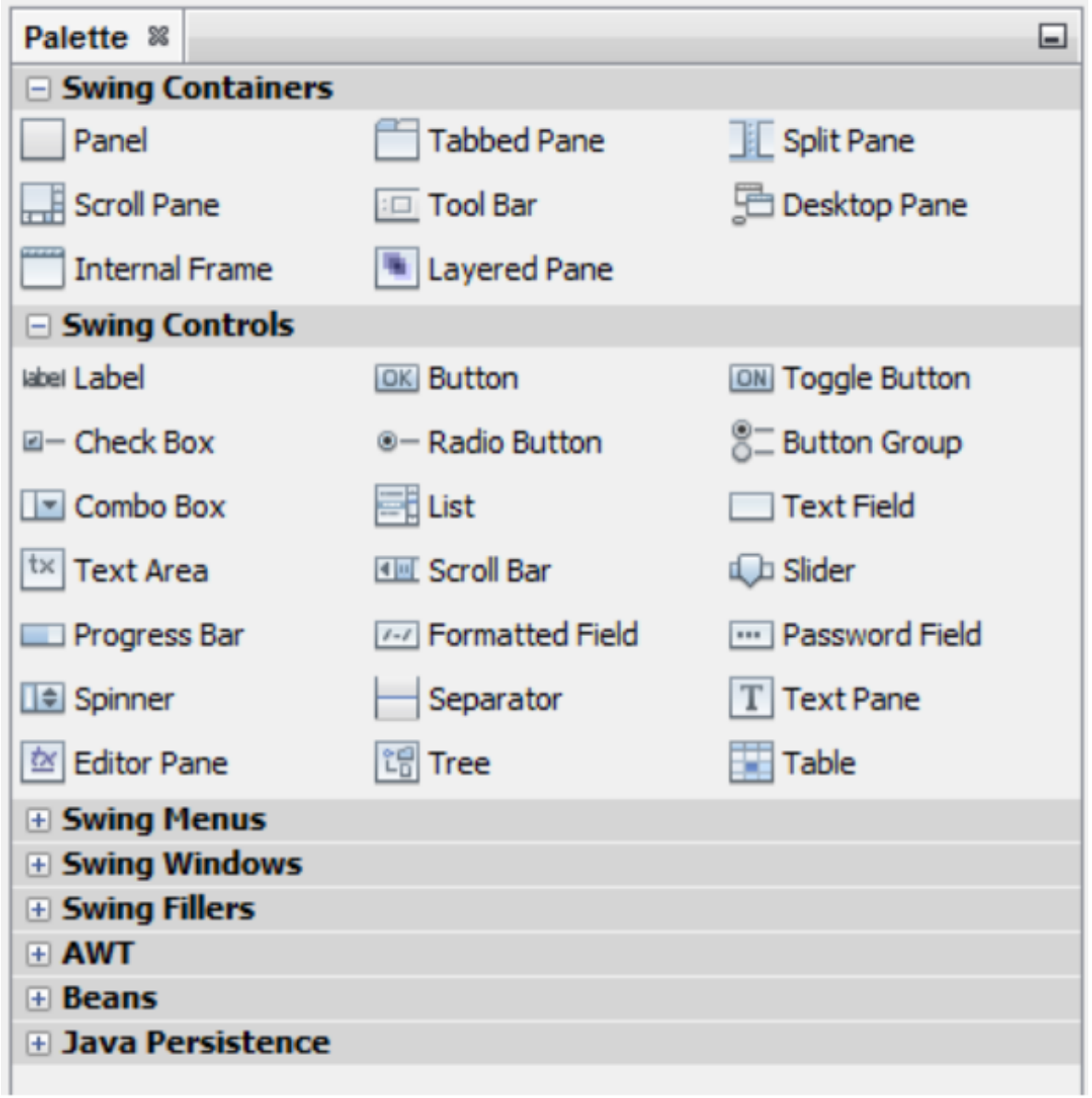
Java ME

- A plataforma JME contém configurações e bibliotecas trabalhadas especialmente para a atuação em dispositivos portáteis.
  - Assim, o desenvolvedor tem maior facilidade para lidar com as limitações de processamento e memória, por exemplo.
- Um exemplo, é a configuração chamada CLDC (Connected Limited Device Configuration), destinada a dispositivos com recursos de hardware bastante limitados, como processadores de 16 bits e memórias com 512 KB de capacidade.

Tecnologias Java
AWT e Swing

- São 2 toolkits de desenvolvimento gráfico do Java:
  - AWT (Abstract Window Toolkit) – Primeiro toolkit que existiu no Java.
    - Componentes têm aparência dependente do sistema operacional
  - Swing – Toolkit que estende/substitui o AWT:
    - Componentes são independentes da plataforma nativa.

Tecnologias Java
AWT e Swing



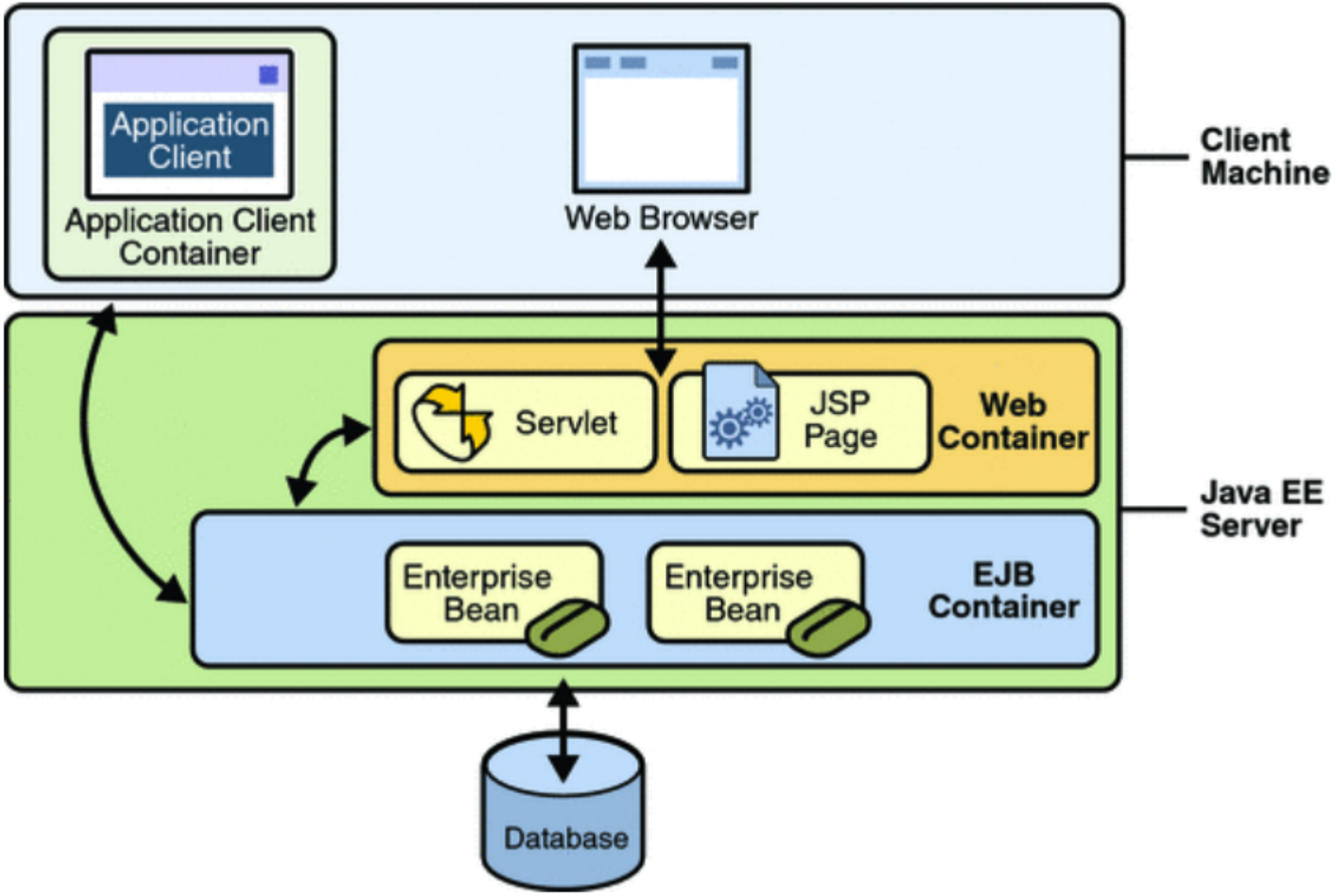
Tecnologias Java
JDBC

- JDBC (Java Database Connectivity) é um conjunto de classes e interfaces (APIs) escritas em Java que fazem o envio de instruções SQL para qualquer banco de dados relacional.
- Possibilita o uso de bancos de dados já instalados.

Tecnologias Java
EJB

- EJB (Enterprise Java Beans) é um componente do tipo servidor que executa no container do servidor de aplicação.
- Os principais objetivos da tecnologia EJB são fornecer um rápido e simplificado desenvolvimento de aplicações Java baseado em componentes distribuídos, transacionais, seguros e portáveis.

Tecnologias Java
EJB



https://docs.oracle.com/javaee/5/tutorial/doc/bnbyl.html

Tecnologias Java
JAR

- É um arquivo compactado usado para distribuir um conjunto de classes Java, um aplicativo java, ou outros itens como imagens, arquivos XMLs, entre outros.
- É usado para armazenar classes compiladas e metadados associados que podem constituir um programa.

Ambientes de desenvolvimento

- Ambientes de desenvolvimento integrado (IDE – *Integrated Development Environment*) oferecem facilidades para codificar em determinada linguagem de programação.
- Exemplos de IDEs:
  - Netbeans;
  - Eclipse;
  - IntelliJ.

Ambientes de desenvolvimento

- Principais características:
  - Suporte a projetos desktop, web e mobile, além de outras linguagens de programação;
  - Possibilidade de desenvolvimento e/ou instalação de plugins de terceiros;
  - Personalização de editores, visões, refatoração, CVS, etc.

Considerações finais

- Java oferece uma API abrangente para diferentes domínios de programação.
  - Exemplos: Sistemas desktop, sistemas web, sistemas embarcados.
- Não existe a melhor IDE;
- A escolha deve considerar a afinidade da equipe.

Referências

- HORSTMANN, CAY S. e CORNELL, GARY: Core Java, V.1 - Fundamentos, 8a. edição, Pearson, 2010.
- SIERRA, KATHY e BATES, BERT: Use A Cabeça! – Java, 2ª. Edição, Alta Books, 2008.
- DEITEL, HARVEY M.: Java - Como Programar, 8ª ed, Prentice Hall Brasil, 2010.