INTRODUÇÃO AO JAVA

Prof. Gustavo Wagner

Java Básico

IESP-PB

JAVA

- Criada pela Sun Microsystems
- Especificação de uma linguagem
  - programação orientada a objetos
  - propósito geral
  - Hoje padronização no Java Community Process (JCP)
- Normalmente associada à explosão da Web como ambiente de trabalho e lazer (applets)
- Exemplo: Applet Troca de Mensagens

JAVA

- Ao mesmo tempo uma linguagem e uma plataforma (máquina virtual e bibliotecas)
- Surgiu do questionamento Eficiência vs. Portabilidade (simplicidade)
- Idéias inovadoras que influenciaram a indústria de forma significativa
  - Visual J++, C#

HISTÓRICO

- 1990: início dos trabalhos na Sun com portabilidade
- James Gosling: Pai de Java

  - Não ao software proprietário
- C++ (menos menos)
  - Mais simples
  - Independente de arquitetura

HISTÓRICO

- 1994: Explosão do WWW
  - Tecnologia de navegadores incentivou aquelas idéias
  - Passou de apenas HTML (estático) para aplicações dinâmicas com Applets
- HotJava
  - Mostrou Java pela 1a. Vez (1995)
  - Netscape, IBM, Symantec, Microsoft

HISTÓRICO

- 1996: Primeira versão (1.0)
- 1998: Java 2 (1.2)
- 2003: Java 2 (1.4)
- 2005: Java 2 (1.5) – JDK 5.0
- 2006: Java 2 (1.6) - JDK 6.0 (Mustang)
- 2008: Vendida para a Oracle

AS PALAVRAS CHAVE DE JAVA

- Java é
  - Simples
  - Orientada a objetos
  - Robusta
  - Segura
  - Portável

JAVA É SIMPLES

- Sintaxe familiar a vários programadores (baseada em C e C++)
- Elimina conceitos complexos de C++, tais como: aritmética de ponteiros e gerência de memória;
- Simples, pelo poder oferecido
- Tamanho reduzido

JAVA É ORIENTADA A OBJETOS

- Objetos e Classes
- Encapsulamento de dados e operações
- Troca de mensagens entre objetos
- Subtipos e Herança
- Polimorfismo
- Eliminou complexidades de objetos do C++

JAVA É ROBUSTA

- Ausência de ponteiros: mentira!
  - Prova: NullPointerException! :)
- Fortemente tipada
  - Compilador "chato"
  - A partir de Java 1.5 checa-se inclusive tipo de objetos em tempo de compilação das estruturas de dados
- Coleta de lixo automática: Garbage Collector
- Acesso a `arrays` é verificado (não se pode acessar index fora da faixa)
- Variáveis são iniciadas automaticamente (com exceção de variáveis locais)

JAVA É SEGURA

- Ambientes distribuídos demandam esta característica
- Série de restrições de segurança
- Mecanismo de segurança de applets
- Processo de melhoria constante
  - Problemas existem

JAVA É PORTÁVEL

- Independência de plataforma...
- Em tese, redução de custos com migração, treinamento...
- Exemplo: Tamanho de tipos de dados (em Java o int tem 4 bytes, independentemente do SO)
- Bibliotecas funcionam com relativa independência
  - Contra-exemplo: AWT

JAVA É INDEPENDENTE DE PLATAFORMA

- Característica chave de Java
- Principal fator de inovação da plataforma Java
  - Quase todas as decisões dos projetistas de Java priorizam esta característica
- Merece mais detalhes...

COMPILAÇÃO X INTERPRETAÇÃO

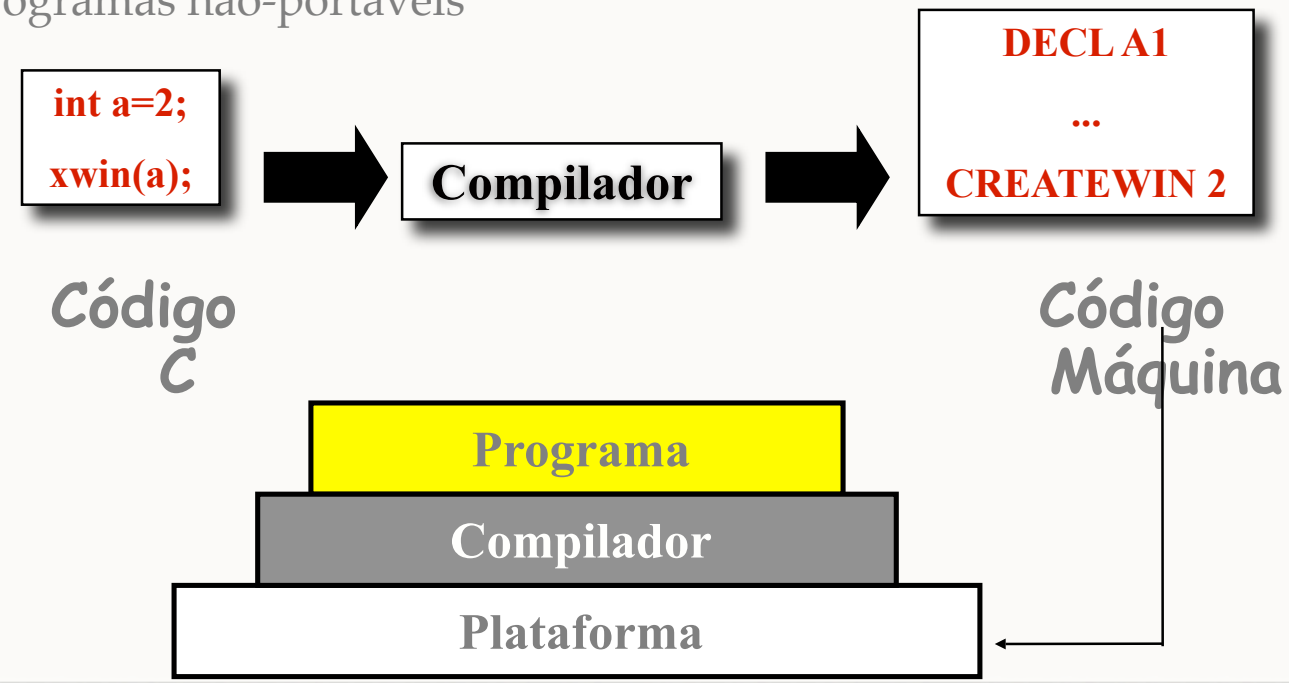
- Compilação
  - Código dependente de máquina é gerado a partir de um código-fonte
- Interpretação
  - Código-fonte é executado diretamente sem a geração de código dependente de máquina

PLATAFORMAS

- Plataforma = Sistema Computacional + Sistema Operacional
- Normalmente nomeada pelo sistema operacional
  - Ex: Windows, Linux (Unix), Mac
- Diferem no conjunto de instruções

PROGRAMAS E PLATAFORMAS

- Programas são compilados (traduzidos) para linguagem de máquina
  - Uso de bibliotecas (API) específicas da plataforma
  - Programas não-portáveis



SISTEMAS MULTIPLATAFORMA

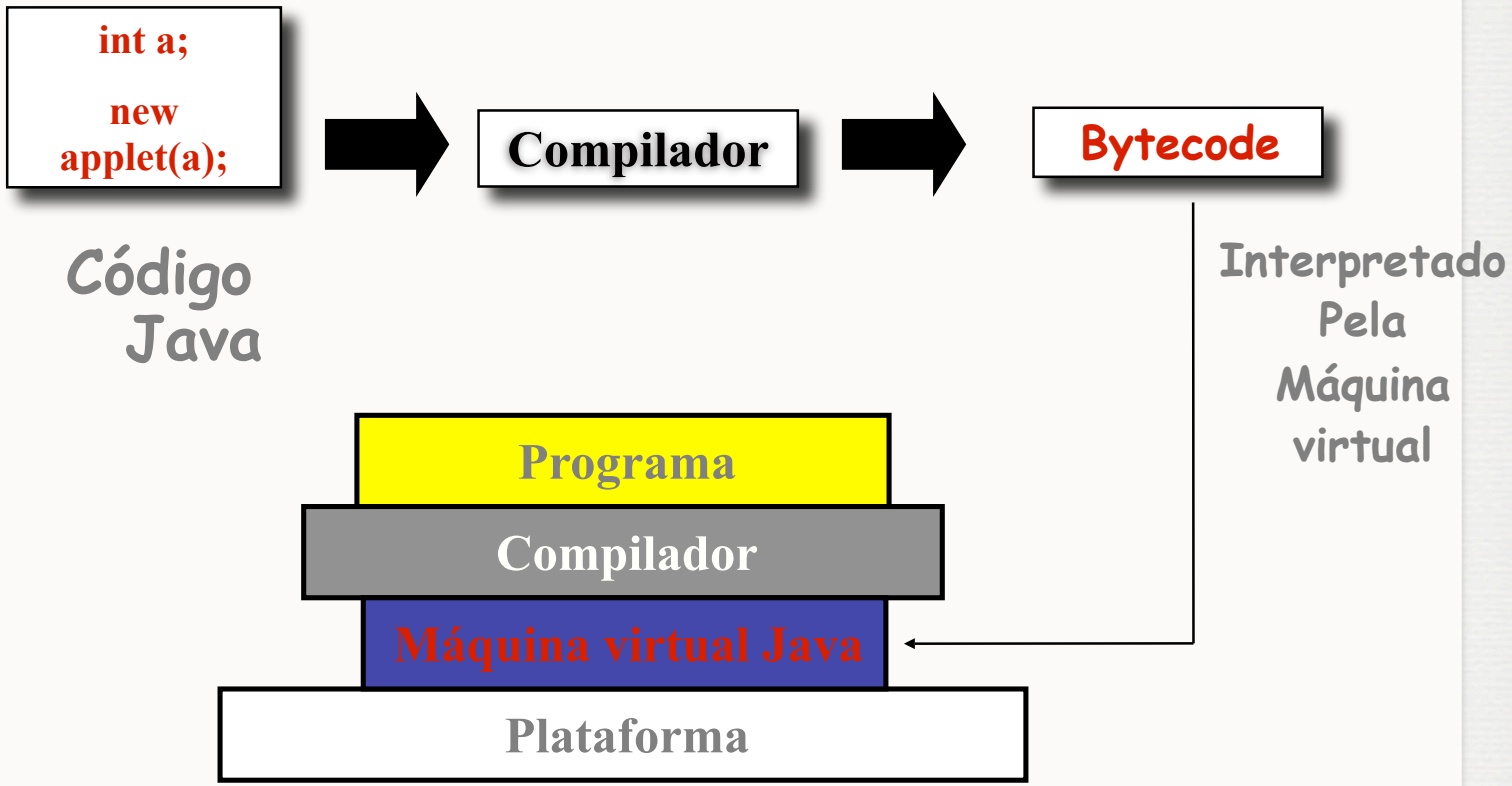
- Portabilidade
  - Write Once, Run Anywhere
  - Importante em aplicações distribuídas (Internet)
- Um único programa executa uniformemente em "qualquer" plataforma

JAVA!!

JAVA É MULTIPLATAFORMA

- Combinação compilação + interpretação
- Máquina virtual Java: ponte entre programas e plataforma real
  - "plataforma" Java
- Compilador Java traduz programas para `bytecodes` que são interpretados para a plataforma real

JAVA É MULTIPLATAFORMA



ENGENHARIA REVERSA

- O bytecode, por ser interpretado, pode ser “descompilado”;
- Código fonte pode ser recuperado, mesmo depois da compilação;
- Solução: `Obfuscator`;

VANTAGENS DE SISTEMAS MULTIPLATAFORMAS

- Não há necessidade de mudanças em programas para funcionamento em diferentes plataformas
- Apenas uma versão do programa é suficiente para distribuição multiplataforma
  - Software para Internet

DESVANTAGENS DE SISTEMAS MULTIPLATAFORMAS

- Perde-se poder pela renúncia a algumas instruções particulares
  - Denominador comum
- Interpretação pode ter desempenho pior do que compilação
  - Em particular, a plataforma Java evoluiu bastante
  - Just-in-time (JIT)

EVOLUÇÃO DO COMPILADOR JAVA

- Aprendizado com o status da máquina: HotSpot;