

Introdução à Tecnologia Java

Orientação a Objetos - DCC025

Prof. Edmar Welington Oliveira
edmar.oliveira@ufjf.edu.br

Universidade Federal de Juiz de Fora - UFJF
Departamento de Ciência da Computação - DCC

Tecnologia Java

- "Java" é usado para referir-se a
 - Uma linguagem de programação orientada a objetos
 - Uma coleção de APIs* para o desenvolvimento de aplicações
 - Um ambiente de execução presente em
 - Browsers, SOs, celulares, cartões inteligentes, eletrodomésticos, etc.

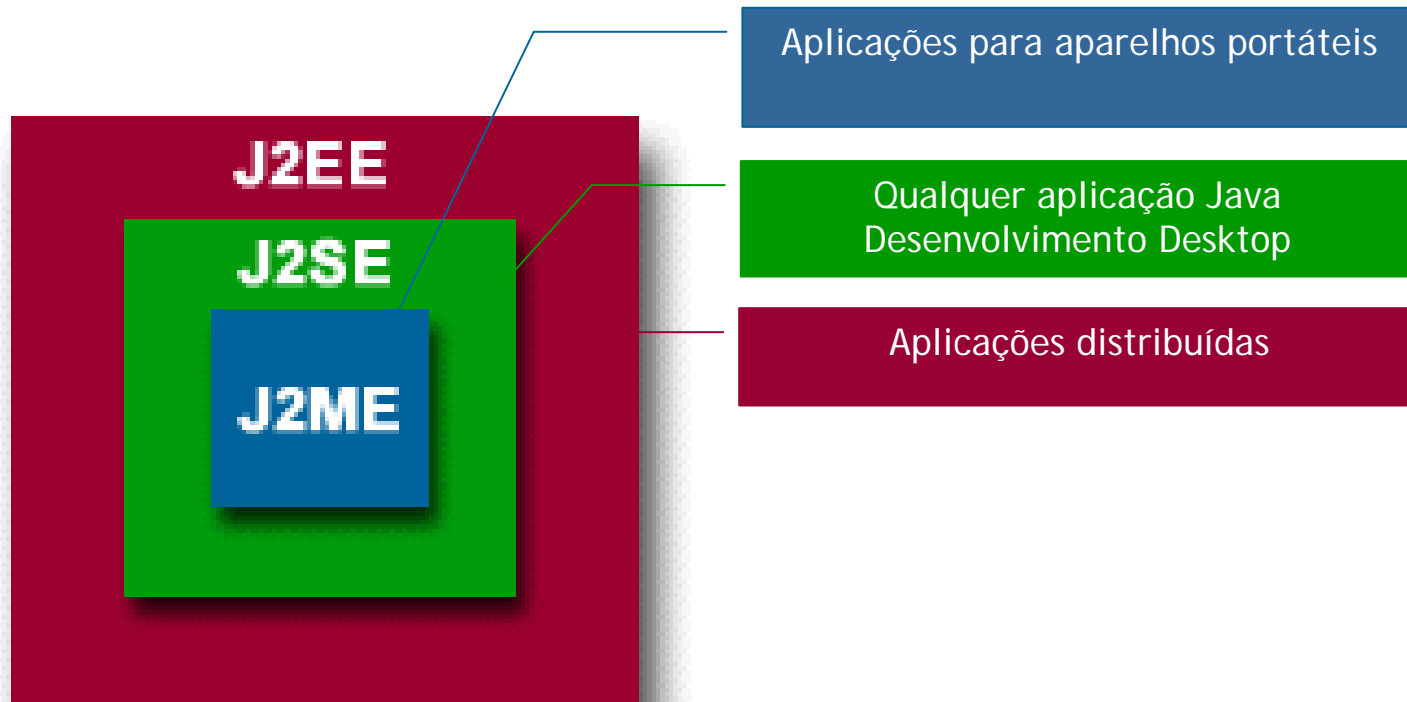
- API: Application Programming Interface

Tecnologia Java

- Coleção de APIs

- Possui uma coleção de APIs (Bibliotecas) para construção de aplicações
- Utilizadas pelos ambientes de execução (JRE) e desenvolvimento (SDK)
 - **JRE: Java Runtime Environment:** Engloba tudo o que é necessário para executar aplicações Java
 - **SDK: Software Development Kit:** Coleção de ferramentas para, entre outras tarefas, compilar, executar e depurar aplicações Java.

Plataformas de Desenvolvimento



The Java Family

Cada uma dessas siglas engloba um conjunto de ferramentas e APIs para suporte à desenvolvimento Java. Contudo, cada uma se concentra em um nicho de desenvolvimento específico

Linguagem de Programação Java

■ Breve Resumo

- Java é uma linguagem de programação desenvolvida pela Sun Microsystems e lançada em 1995. O seu desenvolvimento foi iniciado em 1991, visando o mercado de bens eletrônicos de consumo.
- Foi projetada desde o início para ser **independente de hardware**, uma vez que as características dos equipamentos variam amplamente neste nicho de desenvolvimento
- Outro objetivo estabelecido desde sua concepção foi o de ser uma **linguagem segura**. Segura tanto no sentido de evitar algumas **falhas comuns** que os programadores costumam cometer durante o desenvolvimento, como no sentido de evitar **ataques externos**

Linguagem Java

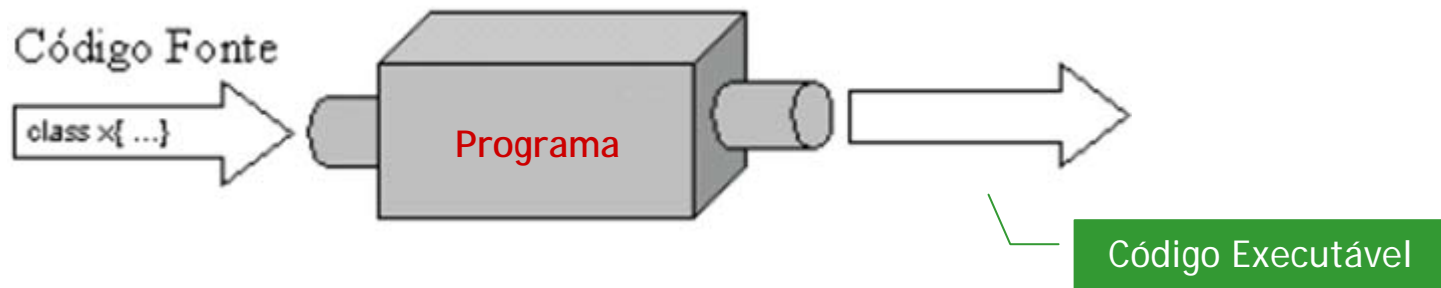
- Breve Resumo

- Utilização em larga escala: Internet

- A Internet também é um ambiente constituído por equipamentos de diferentes arquiteturas e necessita muito de uma linguagem que permita a construção de aplicativos seguros.

Linguagem Java

- Código Interpretável e portátil
- Linguagens de programação podem ser tanto **compiladas** como **interpretadas**. Quando se utiliza uma **linguagem compilada**, é necessário executar um programa para traduzir os arquivos fonte, legíveis em linguagem de alto nível, em código executável.



Linguagem Java

- Código Interpretável e portátil
 - As linguagens compiladas têm a vantagem de **produzir código** de alta performance, o qual está **ajustado** para o funcionamento em um tipo específico de **processador** ou arquitetura de processador.
 - Aplicativos compilados, chamados de código binário, **só podem rodar no tipo de computador para o qual foram compilados**, uma vez que esses aplicativos consistem, na realidade, em instruções em linguagem de máquina, entendidas e executadas pelo microprocessador.

Linguagem Java

- Linguagem Interpretadas

- Programas escritos em linguagens interpretadas não são convertidos em um arquivo executável. Eles são executados utilizando um outro programa, o interpretador, que lê o código-fonte e o interpreta diretamente, durante a sua execução. O texto do programa é traduzido na medida em que vai sendo executado
- Programas **interpretados são geralmente mais lentos do que os compilados**, mas são também geralmente mais flexíveis, já que podem interagir com o ambiente mais facilmente

Linguagem Java

- Compilação e Interpretação

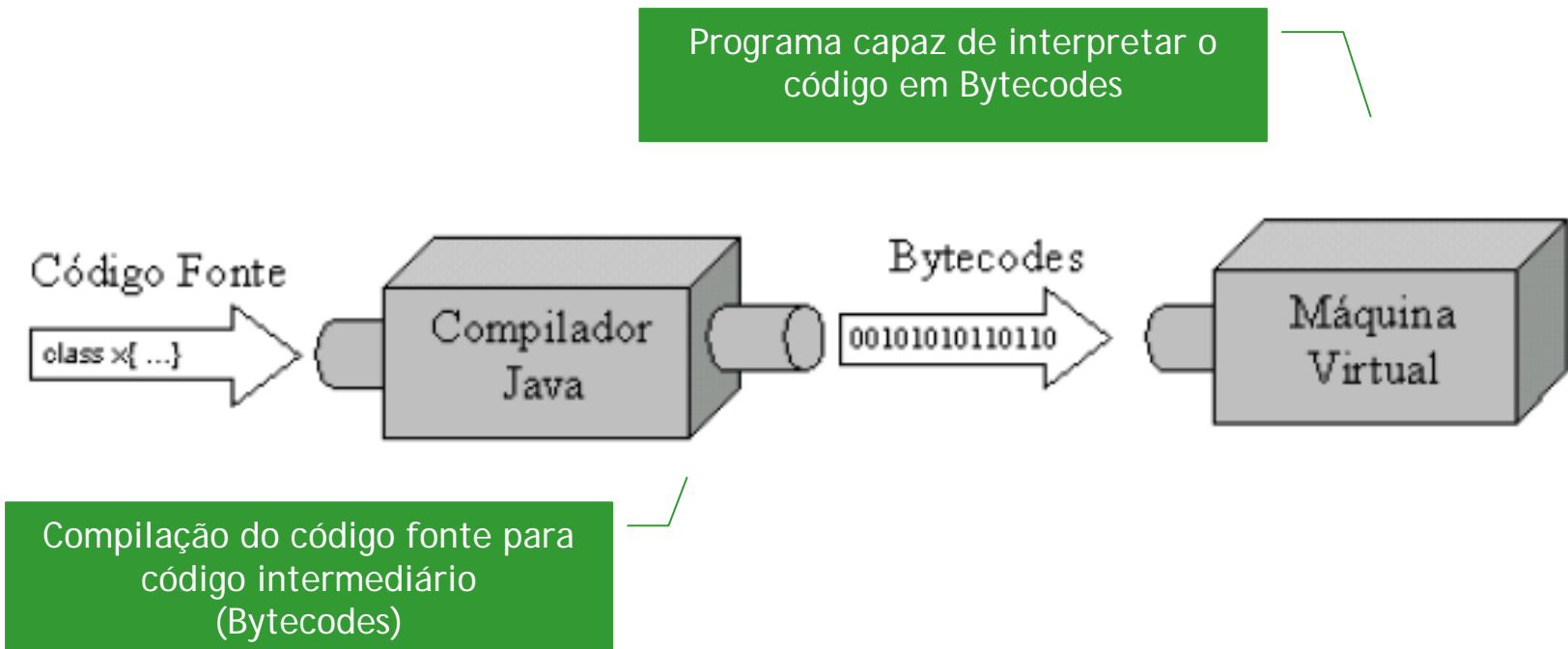
- A linguagem Java é tanto **compilada** como **interpretada**.
- Após escrever um programa em Java, utilizando um editor de textos qualquer, salva-se o programa como código fonte. A seguir, pode-se compilar esse fonte, a fim de produzir um tipo de arquivo binário chamado de arquivo de classe. Esses arquivos não são executados diretamente pois eles não contêm instruções que são entendidas diretamente pelos processadores

Linguagem Java

■ Continuação

- Os programas Java são compilados em um **formato intermediário** chamado **bytecodes**. Assim, esses programas podem ser executados em qualquer sistema através de um interpretador Java (Java Runtime Environment - JRE).
- Vantagem
 - O código precisa ser escrito e compilado apenas uma vez, pois os bytecodes gerados serão executados da mesma forma em qualquer plataforma de hardware e software.

Linguagem Java



Linguagem Java

■ Segurança

- Por ter seu projeto voltado para a simplicidade de código, as possibilidades de erro de programação em Java são reduzidas. O processo de compilação é projetado para a detecção prévia dos possíveis erros, evitando que os erros se manifestem em tempo de execução
- Além disso
 - O uso de código para tratamento de exceções · exception handling - permite manter a consistência da aplicação no caso de erros

Linguagem Java

- Aplicações Distribuídas

- Com a larga utilização dos recursos da rede Internet, a linguagem teve seu projeto voltado para as aplicações em rede. Assim, a linguagem traz classes para o suporte a vários níveis de conectividade: acesso a URLs (padrão Internet), uso de conexões em sockets, criação de protocolos, criação de clientes e servidores.

Linguagem Java

- Processamento Paralelo

- Para permitir uma melhor performance de execução, mesmo em tarefas de maior complexidade, a linguagem permite a **programação de threads** - processos de execução de tarefas que ocorrem simultaneamente. A linguagem traz também mecanismos para sincronização, ativação e desativação parametrizada desses processos.

Desenvolvimento Java

- Recursos para desenvolvimento
 - A Sun Microsystems preocupou-se em fornecer aos futuros desenvolvedores de aplicações Java um pacote de ferramentas e bibliotecas básicas. Esse pacote, chamado Java Development Kit (JDK)
 - O JDK oferece ao desenvolvedor uma série de ferramentas, como:
 - Interpretador
 - Compilador
 - Debugger
 - Gerador de documentação
 - Visualizador de applets
 - Visualizador de bytecodes
 - Compactador de classes
 - Etc..

Linguagem Java

- JDK - ferramentas mais utilizadas

- Javac

- É o compilador da linguagem Java. Ele lê arquivos fonte .java e gera arquivos de classe .class no formato de bytecodes. Para cada classe especificada é gerado um arquivo de classe chamado **NomedaClasse.class**.
 - **Importante:** o arquivo fonte deve ter o nome da classe que ele define seguido da extensão .java, caso contrário o compilador acusa erro.

Linguagem Java

- JDK - ferramentas mais utilizadas

- Java

- O interpretador java é utilizado para executar aplicações em Java. Ele interpreta os bytecodes gerados pelo javac. O interpretador é executado a partir de linha de comando da forma:

- `java NomedaClasse arg1 arg2 arg3...`

- Ao ser executado, o interpretador busca o arquivo `NomedaClasse.class`, nele procura um `método public static void main()` e a ele passa os argumentos que recebeu (ou não, pois os argumentos são opcionais e sua existência depende da funcionalidade da classe que está sendo interpretada).

Processo Completo

HelloWorld - Notepad

```
File Edit Format View Help
public class Helloworld
{
    public static void main (String args[])
    {
        System.out.println("oi Mundo");
    }
}
```

HelloWorld.java

Código Fonte

Compilação: javac

C:\WINDOWS\system32\cmd.exe

```
F:\eclipse\workspace\Aula\src>javac HelloWorld.java
F:\eclipse\workspace\Aula\src>_
```

HelloWorld.class

Interpretação: java

C:\WINDOWS\system32\cmd.exe

```
F:\eclipse\workspace\Aula\src>java HelloWorld
Oi Mundo
F:\eclipse\workspace\Aula\src>_
```

Linguagem Java

- API Java
 - O conjunto de bibliotecas inclusas no JDK é conhecido como Java **API (Interface de Programação de Aplicações)**. Nessas bibliotecas estão um conjunto grande de classes, organizadas em pacotes. Cada um desses pacotes traz classes com funcionalidade básica e vital para um determinado ramo de programação Java.

Programação Java

Exemplo de Programa Java

```
/** Aplicação Hello World */  
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, world!");  
    }  
}
```

Esta mini-aplicação em Java imprime um texto na tela quando executada via linha de comando

Anatomia de Programa Java

Comentário de bloco

Nome da classe

Nome do método

Declaração de argumento

variável local: args
tipo: String[]

Ponto-e-vírgula
é obrigatório no
final de toda
instrução

Definição de método main()

Atribuição de argumento
para o método println()

Definição de classe
HelloWorld

Chamada de método println()
via objeto out acessível
através da classe System

```
/** Aplicação Hello World */
```

```
public class HelloWorld {
```

```
    public static void main(String[] args) {
```

```
        System.out.println("Hello, world!");
```

```
    }
```

```
}
```

Programação Java

Nome de Classe

```
public class OlaMundo
{
    public void exibeOla()
    {
        System.out.println("Ola, Mundo!");
    }

    public static void main(String args[])
    {
        OlaMundo obj = new OlaMundo();
        Obj.exibeOla();
    }
}
```


Programação Java

```
public class OlaMundo
```

```
{
```

```
    public void exibeOla()
```

```
    {
```

```
        System.out.println("Ola, Mundo!");
```

```
    }
```

```
    public static void main(String args[])
```

```
    {
```

```
        OlaMundo obj = new OlaMundo();
```

```
        obj.exibeOla();
```

```
    }
```

```
}
```

Definição de 2 métodos

OBS: Não é possível fazer um programa Java sem recorrer às classes, uma vez que os procedimentos são definidos como métodos de classes.

Programação Java

```
public class OlaMundo
{
    public void exibeOla()
    {
        System.out.println("Ola, Mundo!");
    }

    public static void main(String args[])
    {
        OlaMundo obj = new OlaMundo();
        obj.exibeOla();
    }
}
```

Os métodos com o nome **main** são métodos especiais e servem como **ponto inicial para execução do programa**.

Ou seja, a execução do programa será iniciada a partir da execução de um método `main()`.

Programação Java

```
public class OlaMundo
{
    public void exibeOla()
    {
        System.out.println("Ola, Mundo!");
    }

    public static void main(String args[])
    {
        OlaMundo obj = new OlaMundo();
        obj.exibeOla();
    }
}
```

Cria objeto de classe

Invocação de método

Programação Java

```
public class OlaMundo
{
    public void exibeOla()
    {
        System.out.println("Ola, Mundo!");
    }

    public static void main(String args[])
    {
        OlaMundo obj = new OlaMundo();
        Obj.exibeOla();
    }
}
```

Este programa deve ser salvo em um arquivo contendo **O MESMO NOME DA CLASSE** e com as mesmas letras maiúsculas e minúsculas e com a extensão ".java". Portanto o arquivo contendo o programa acima deverá se chamar **OlaMundo.java**

Programação Java

```
public class OlaMundo
{
    public void exibeOla()
    {
        System.out.println("Ola, Mundo!");
    }

    public static void main(String args[])
    {
        OlaMundo obj = new OlaMundo();
        Obj.exibeOla();
    }
}
```

Compilação

Para compilar o programa basta digitar o comando: **javac OlaMundo.java**

O código Java é traduzido para instruções de uma máquina virtual (bytecodes) para que possa ser executado de forma independente da plataforma. O código em bytecodes é armazenado em um arquivo com o mesmo nome do original e com a extensão ".class".

Após compilar, será gerado o arquivo:
OlaMundo.class

Programação Java

```
public class OlaMundo
{
    public void exibeOla()
    {
        System.out.println("Ola, Mundo!");
    }

    public static void main(String args[])
    {
        OlaMundo obj = new OlaMundo();
        Obj.exibeOla();
    }
}
```

Execução

A invocação da máquina virtual para executar o código em bytecodes é efetuada por meio do seguinte comando: **java OlaMundo**

Quando recebe um arquivo de bytecodes para a executar, a **máquina virtual procura o método main() em uma classe com o mesmo nome do arquivo**. Uma vez encontrado o método a execução é iniciada com a execução do método.

Introdução à Tecnologia Java

Orientação a Objetos - DCC025

Prof. Edmar Welington Oliveira
edmar.oliveira@ufjf.edu.br

Universidade Federal de Juiz de Fora - UFJF
Departamento de Ciência da Computação - DCC