

Lógica de Programação

Técnico em Desenvolvimento de Sistemas

JAVA

O QUE É JAVA?

- Java é uma linguagem de programação orientada a objetos desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems.
- Em 2008 o Java foi adquirido pela empresa Oracle Corporation.

O QUE É JAVA?

- A página principal do Java é:

<http://www.oracle.com/technetwork/java/>



O QUE É JAVA?

- A sintaxe utilizada deriva do C++ porém com um modelo mais simples.
- Como sua principal característica, todo o código é escrito dentro de uma classe e tudo é um objeto, com exceção dos tipos intrínsecos (números reais e ordinais, valores booleanos e caracteres) que não são classes por questões da funcionalidade exercida.

O QUE É JAVA?

- Objetivos principais:
- Simples
- Orientada a objetos
- Fácil aprendizagem não somente para programadores experientes.

PORQUE QUE JAVA?

- Arquitetura neutra e portátil
- Multiplataforma
- Alta performance
- Segurança e solidez
- Multi dispositivos
- WEB e Desktop

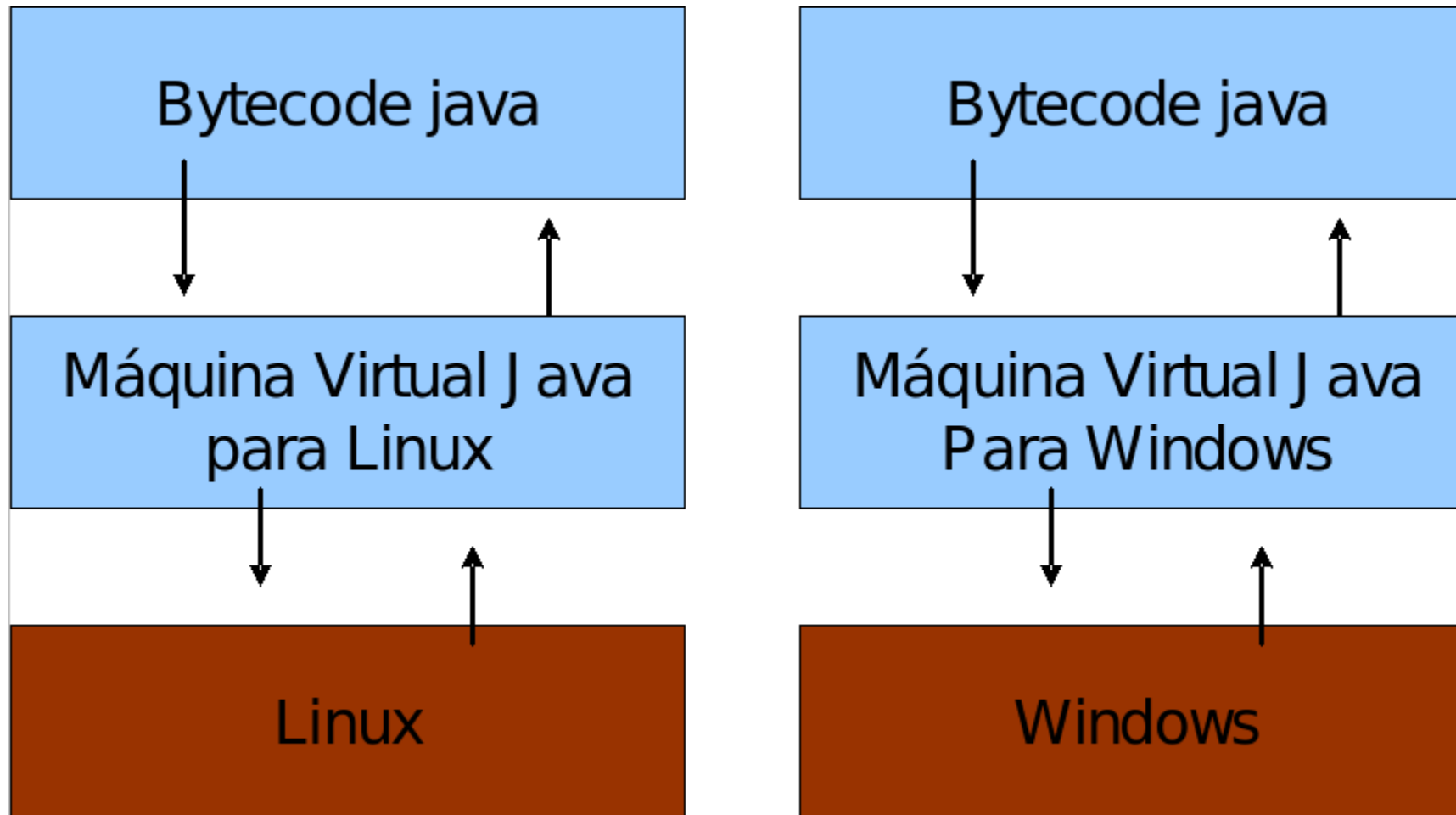
PORQUE QUE JAVA?

- Muitas empresas de software disponibilizam programas feitos nessa linguagem, tornando a instalação da Máquina Virtual Java(JVM) quase que obrigatória na maioria dos sistema operacionais atuais.
- Entre os grandes exemplos do uso da tecnologia está o acesso a serviços bancários online e os vários aplicativos da Receita Federal brasileira, como o programa de declaração de imposto de renda.

JVM?

- Entre o sistema operacional e a aplicação existe uma camada extra responsável por "traduzir" o que a aplicação deseja fazer para as respectivas chamadas do sistema operacional onde ela está rodando no momento:

JVM?



JVM?

- Essa camada não entende código java, mas sim um código de máquina específico.
- Esse código de máquina é gerado por um compilador java, como o javac, e é conhecido por "bytecode".

JVM?

- O compilador Java gera esse bytecode que, diferente das linguagens sem máquina virtual, vai servir para diferentes sistemas operacionais, já que ele vai ser "traduzido" pela JVM.

ESTRUTURA DO JAVA

Estrutura dos programas simples em Java

```
Import java.io.* //Exemplo de import  
  
Public class nomeclasse{  
    Public static void main(String args[]){  
        //Código para executar os programas  
    }//fecha main  
}//fecha classe
```

Obs.: o import que está no início do programa é utilizado para importar classes de bibliotecas prontas do Java. Existem várias bibliotecas, a utilizada no exemplo acima trata de entrada e saída do programa (Leitura e escrita).

ESTRUTURA DO JAVA

- Case Sensitive (distinção entre maiúsculas e minúsculas)
- Não usar acentuação(´,`,~,^,ç)
- Final de linha de instrução colocar ;

ARQUIVO FONTE (Source File)

- Em java, cada classe(class) é colocada em um arquivo que representa parte de uma aplicação ou toda a aplicação(no caso de programas muito pequenos).
- São gerados com a extensão .java e podem ser facilmente criados utilizando o notepad por exemplo.
- Esses arquivos **DEVEM** possuir o mesmo nome da classe que está em seu código.

ARQUIVO FONTE (Source File)

Nome	Data de modificaç...	Tipo	Tamanho
 Candidato	02/05/2012 23:31	Arquivo JAVA	1 KB
 ControladorEleitor	03/05/2012 11:27	Arquivo JAVA	3 KB
 ControladorMesario	03/05/2012 11:31	Arquivo JAVA	1 KB
 Debug	02/05/2012 23:26	Arquivo JAVA	2 KB
 Eleitor	03/05/2012 00:15	Arquivo JAVA	1 KB
 ListaCandidato	02/05/2012 22:38	Arquivo JAVA	3 KB
 ListaEleitor	03/05/2012 20:02	Arquivo JAVA	5 KB
 Principal	03/05/2012 00:23	Arquivo JAVA	2 KB
 Relatorio	03/05/2012 11:31	Arquivo JAVA	2 KB
 TelaDebug	02/05/2012 23:19	Arquivo JAVA	2 KB
 TelaEleitor	03/05/2012 11:27	Arquivo JAVA	2 KB
 TelaMesario	02/05/2012 22:25	Arquivo JAVA	2 KB
 TelaPrincipal	03/05/2012 00:38	Arquivo JAVA	2 KB

CLASSE(Class)

- Em uma classe java são colocados os métodos (methods), funções ou procedimentos.
- Todo o código deve estar em alguma classe, pois quando executamos algum aplicativo java nós estamos, na verdade, executando uma classe.

CLASSE(Class)

- Diferentemente de um arquivo fonte (Source File) que só pode conter uma classe, uma classe pode conter vários métodos.
- Deve estar em um Arquivo Fonte (Source File)
- Deve ser declarada e seguida de {, indicando seu início
- Deve ser encerrada com um }

CLASSE(Class)

- O código de uma Classe fica entre um par de chaves { }, são entre essas chaves que serão colocados os métodos.
- Lembrando que uma classe sempre inicia com letra maiúscula.

CLASSE(Class)

Listagem 1: Exemplo de uma classe contida em um Source File MyClass.java

```
1 public class MyClass{  
2 // código vai aqui  
3 }
```

MÉTODOS(Methods)

- Os métodos, funções ou procedimentos são onde declararemos o código das nossas aplicações java.
- Assim como classes, os métodos devem ser escritos acompanhados de um par de chaves “{}”.
- Lembrando que um método sempre inicia com letra minúscula.

MÉTODOS(Methods)

Listagem 2: Exemplo de um método contido em uma Classe

```
1 public class MyClass{  
2     public void meuMetodo(/*argumentos*/){  
3     }  
4 }
```

O MÉTODO main

- Quando o java virtual machine(JVM) inicia, ele procura na sua classe principal por um método muito específico, chamado de método main.
- Uma aplicação java obrigatoriamente deverá possuir pelo menos uma classe e um método main.
- É por esse método main que o JVM começará a executar.

O MÉTODO main

- Por regra , todo método main deverá ser : Público(public), estático(static), sem retorno(void), com nome de “main”, e deverá receber como argumento um array do tipo String.

Listagem 5: Exemplo de um método main contido em uma Classe

```
1 public class MyClass{  
2     public static void main(String[] args){  
3     }  
4 }
```


COMENTÁRIOS

Para fazer um comentário, você pode usar o `//` para comentar até o final da linha, ou então usar o `/* */` para comentar o que estiver entre eles.

```
/* comentário daqui,  
ate aqui */
```

```
// uma linha de comentário sobre a idade  
int idade;
```

AMBIENTE JAVA

- O JDK (Java Development Kit) possui todo o ambiente necessário para desenvolver e executar aplicativos em Java.
 - Esse Kit de desenvolvimento é composto pela Java Virtual Machine (JVM), compilador da linguagem e uma série de códigos (bibliotecas) que nos auxiliam no desenvolvimento de aplicações.
- O JRE (Java Runtime Environment) é o ambiente de execução Java, é o mínimo que você precisa ter instalado para poder rodar um aplicativo Java.

VARIÁVEIS DE AMBIENTE

- Guardam informações sobre o ambiente atual, tanto pode ser do console/terminal em uso, como do usuário ou do sistema.
- São informações que dizem aos programas e aplicativos alguns dados básicos para que possam executar determinadas “ações”.

VARIÁVEIS DE AMBIENTE JAVA

- São 3:

1. JAVA_HOME

2. CLASSPATH

3. Path

VARIÁVEIS DE AMBIENTE JAVA

- Para criá-las:

1. Abra o PAINEL DE CONTROLE

2. SISTEMA E SEGURANÇA

3. SISTEMA

4. CONFIGURAÇÕES AVANÇADAS DO SISTEMA

5. VARIÁVEIS DE AMBIENTE

JAVA_HOME

- Clique no botão “Nova” em “Variáveis do sistema”;
- Nome da variável: JAVA_HOME
- Valor da variável: coloque aqui o endereço de instalação
C:\Program Files\Java\jdk1.8.0_45
- Clique em OK

CLASSPATH

- Clique no botão “Nova” em “Variáveis do sistema”;
- Nome da variável: **CLASSPATH**
- Valor da variável: **%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\htmlconverter.jar;%JAVA_HOME%\jre\lib;%JAVA_HOME%\jre\lib\rt.jar;**
- Clique em OK

PATH

- Selecione a variável Path em Variáveis do sistema.
- Clique no botão Editar.
- Clique no botão Novo.
- Valor da variável: **%JAVA_HOME%\bin**
- Clique em OK

COMPILANDO O PRIMEIRO PROGRAMA

- Impressão de uma linha
 - Para mostrar uma linha, podemos fazer:
 - `System.out.println("Minha primeira aplicação Java!");`

COMPILANDO O PRIMEIRO PROGRAMA

- Abra o bloco de notas e digite o código:
- ```
class MeuPrograma{
public static void main(String[] args){
 //comando q imprime 1 linha com o conteúdo entre aspas
 System.out.println("Minha primeira aplicação Java!");
}
}
```

# COMPILANDO O PRIMEIRO PROGRAMA

- Salve o código em sua pasta de preferencia com o nome MeuPrograma.java
- Abra o Prompt de Comando e navegue até a pasta em que salvou o arquivo fonte.

# COMPILANDO O PRIMEIRO PROGRAMA

- Digite o comando: `javac MeuPrograma.java`
- Se houver algum erro no seu código será listado nesse momento
- Senão será gerado o arquivo compilado `MeuPrograma.class`

# COMPILANDO O PRIMEIRO PROGRAMA

- Digite o comando: `java MeuPrograma`
- Se houver algum erro no seu arquivo compilado será listado nesse momento
- Senão será executado o arquivo compilado e deve ser listada a mensagem: Minha primeira aplicação Java!