

UNIVERSIDADE FEDERAL DA BAHIA  
INSTITUTO DE MATEMÁTICA  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO  
DISCIPLINA: LINGUAGENS PARA APLICAÇÃO COMERCIAL

Java e RichFaces

Salvador-BA

Abril - 2009

ACÁSSIO NOVAIS QUEIROZ

ADSON VINICIUS DAMASCENO

GUSTAVO RAMOS ALMEIDA

Trabalho Semestral - Etapa teórica

Java e RichFaces

Atividade apresentada como requisito para avaliação no curso de Ciência da Computação da Universidade Federal da Bahia na disciplina Linguagens Para Aplicação Comercial orientada pelo professor Adonai Estrela Medrado.

Salvador-BA

## SUMÁRIO

|                                          |    |
|------------------------------------------|----|
| 1. Introdução .....                      | 5  |
| 2. Versões .....                         | 5  |
| 3. Java em concursos públicos .....      | 6  |
| 4. Requisitos mínimos para Java .....    | 9  |
| 5. Sintaxe da Linguagem.....             | 10 |
| Tipos básicos .....                      | 10 |
| Declaração de variáveis.....             | 10 |
| Classes .....                            | 10 |
| Instruções condicionais .....            | 11 |
| Instruções de repetição .....            | 11 |
| for .....                                | 12 |
| for-each .....                           | 12 |
| while .....                              | 12 |
| do while .....                           | 12 |
| Métodos .....                            | 13 |
| Objetos.....                             | 13 |
| Comentários.....                         | 13 |
| Exceções.....                            | 14 |
| 6. Tipo de aplicações .....              | 15 |
| 7. IDE disponíveis .....                 | 15 |
| 8. Padronização ISO/ANSI.....            | 16 |
| 9. Conexão com Banco de Dados.....       | 16 |
| 10. Interação com outras linguagens..... | 17 |
| 11. JavaDoc .....                        | 17 |
| 12. RichFaces.....                       | 18 |

|       |                                                |    |
|-------|------------------------------------------------|----|
| 12.1. | Características .....                          | 19 |
| 12.2. | Licença.....                                   | 19 |
| 12.3. | Sistemas Suportados .....                      | 20 |
| 12.4. | Executar um exemplo simples no Jboss 5.x ..... | 20 |
|       | Referências .....                              | 22 |

## **1. Introdução**

Em 1991 um pequeno grupo de desenvolvedores da Sun Microsystems mudou-se para uma empresa filial com o intuito de desenvolver uma nova tecnologia que integrasse computadores a equipamentos eletrodomésticos. Eles imaginavam que a convergência entre esses aparelhos iria acontecer num futuro próximo. Os mentores do projeto, Patrick Naughton, Mike Sheridan, e James Gosling, não tinham como objetivo criar uma linguagem de programação.

No verão de 1992 emergiu o protótipo \*7 (StarSeven), um controle remoto com interface touchscreen que tinha a habilidade de controlar diversos dispositivos e aplicações. Gosling criou uma nova linguagem de programação para o \*7 e a batizou de Oak. A idéia era boa, mas ainda estava cedo demais para viabilizar o negócio. Permitir ao telespectador interagir com a emissora e com a programação era algo visionário e estava longe do que as empresas de TV a cabo tinham capacidade de entender e comprar.

Com o estouro da internet a equipe percebeu uma nova oportunidade. Não era mais necessário criar uma grande infra-estrutura de rede. A internet era o que eles buscavam: uma grande rede interativa.

A tarefa de adaptar o projeto para a internet foi atribuída a Gosling, e em Janeiro de 1995 foi lançada a nova versão do Oak, batizada de Java.

O estático HTML dos browsers que permitiu a rápida disseminação da tecnologia Java. Pois com ela aplicações poderiam ser executadas dentro dos navegadores, nos Applets Java, e disponibilizado pela internet.

## **2. Versões**

JDK 1.0 - Versão inicial.

JDK 1.1 - "Retooling" do modelo de eventos do AWT, são adicionadas "inner classes" à linguagem e surgem a 8 Dec JavaBeans, JDBC e RMI.

O Java 2 Platform, Standard Edition substituiu o JDK para distinguir a base da plataforma J2EE (Java 2 Platform Enterprise Edition) e J2ME (Java 2 Platform , Micro Edition)[9].

J2SE 1.2 - Suporte ao "Reflection" no código, integração do API gráfico Swing nas classes nucleares, incorporação de compilador JIT na JVM, plugin Java, IDL Jave e framework de "collections".

J2SE 1.3 - Surge com o nome de código "kestrel" e incluindo a JVM "hotspot", PMI, Java Sound, interface de designação e diretoria (JNDI) e arquitetura de depuração de plataforma Java (JPDA).

J2SE 1.4 - Surge com o nome de código "Merlin" e inclui suporte de expressões regulares (inspirado no suporte de expressões regulares existente em Perl), NIO sem bloqueamento, API de "logging", parser de XML e processador de XSLT (JAXP) integrados, incorporação de extensões criptográficas e de segurança.

J2SE 5.0 - Surge com o nome de código "tiger" e inclui genéricos, meta-dados, "autoboxing/unboxing", enumerações e Varargs.

Java SE 6 - Surge com o nome de código "mustang" e inclui suporte de "scripting", suporte para anotações "pluggable" e uma data de melhorias em GUI.

Java SE 7 - Surge com o nome de código "dolphin", ainda em desenvolvimento. A JVM terá suporte para linguagens dinâmicas e será lançada uma nova biblioteca para dar suporte à computação paralela em processadores multi-core.

### **3. Java em concursos públicos**

No concurso público [2] houve algumas questões sobre a linguagem Java:

08. Em relação às linguagens de programação e compiladores, é correto afirmar que:

- A) o compilador gera um código de alto nível a partir de um código de baixo nível.
- B) linguagens compiladas, em tempo de execução, são mais lentas que linguagens interpretadas.
- C) a linguagem Object Pascal, presente no Delphi, é um exemplo de linguagem compilada.
- D) são exemplos de linguagens que não utilizam compiladores: Java e Delphi.

A alternativa D é falsa, pois Java possui um compilador, que transforma o código fonte em bytecode, que será interpretado e executado pela Máquina Virtual Java (JVM). A alternativa correta é a C.

15. Considere o programa em Java a seguir:

```
public class Programa {
    private int maior, menor;
    public Programa(int nummaior, int nummenor) {
        maior = nummaior;
        menor = nummenor;
    }
    public int getFunc() {
        int t = 0;
        while (menor > 0) {
            t = menor;
            menor = maior % menor;
            maior = t;
        }
        return maior;
    }
    public static void main(String[] args) {
        Programa p = new Programa(10, 15);
        System.out.println(p.getFunc());
    }
}
```

De acordo com o código, ao final de sua execução, o mesmo exibirá na tela:

A) 0 B) 15 C) 10 D) 5

A alternativa correta é a D. Deve-se ficar atento que a chamada do método *p.getFunc(10,15)* pode confundir o leitor, uma vez supõe-se que o maior número seja atribuído ao primeiro argumento do método.

16. Um dos pontos fortes da linguagem de programação Java é a sua portabilidade para diversas plataformas de hardware. O principal motivo para isso é

A) a existência de funções de baixo nível ou que se comuniquem diretamente com o sistema operacional, como manipulação de interrupções, por exemplo.

B) o fato de a linguagem Java ser dependente do ambiente de execução, isto é, há um compilador para cada plataforma de hardware.

C) a existência de diversas JVM (Java Virtual Machine), uma para cada plataforma de hardware, onde os bytecodes são interpretados e executados.

D) o fato de a linguagem Java ser desenvolvida para a Internet, de modo que qualquer dispositivo de hardware que tenha acesso à Internet deverá acessar a aplicação.

A alternativa correta é a C. Java não possui funções de baixo nível que se comunicam diretamente com o sistema operacional para justamente possibilitar a portabilidade (A incorreta). Existe apenas um compilador para a plataforma Java (B incorreta), que compila o código fonte em bytecode, e este poderá ser interpretado por qualquer JVM. O fato de Java ter sido desenvolvido para a internet não significa que qualquer dispositivo conectado à grande rede poderá utilizá-la. Para que isso ocorra é necessário que este dispositivo possua uma JVM capaz de interpretar os bytecodes.



No concurso público [3] as seguintes questões exigem conhecimento da linguagem Java:

53. Os métodos Java que não retornam valores devem possuir no parâmetro tipo-de-retorno a palavra

- (A) static.
- (B) public.
- (C) void.
- (D) main.
- (E) string args.

A resposta correta é C, pois o tipo de dado void é usado quando um método não possui como retorno um valor.

54. A seqüência de etapas para implementação de um programa Java é

- (A) interpretação, codificação, execução e compilação.
- (B) codificação, interpretação, compilação e execução.
- (C) interpretação, codificação, compilação e execução.
- (D) codificação, compilação, interpretação e execução.
- (E) compilação, codificação, execução e interpretação.

A alternativa correta é a D. Deve-se primeiro escrever o código (codificação), então compilar este código para um bytecode (compilação). Depois o código compilado será interpretado pela JVM (interpretação) e só então será executado (execução).

#### **4. Requisitos mínimos para Java**

Para executar um programa Java, o computador deve possuir a Java Virtual Machine (JVM) instalada, que pode ser obtida no site da Sun [4]. Esse software é responsável por interpretar os bytecodes gerados pelo compilador Java. Por isso, Java é uma linguagem onde o código fonte é compilado para ser executado numa máquina virtual, tornando-a independente de máquina.

## 5. Sintaxe da Linguagem

### Tipos básicos

Os tipos básicos da linguagem Java são:

- `boolean;`
- `char;`
- `byte;`
- `short;`
- `int;`
- `long;`
- `float;`
- `double;`

### Declaração de variáveis

A declaração de variáveis é feita da seguinte forma:

```
<tipo> <identificador>;
```

Por exemplo:

```
boolean b;
```

```
int idade;
```

A declaração de matrizes é da seguinte forma:

```
<tipo>[][] <identificador>;
```

ou

```
<tipo> <identificador>[][];
```

Sendo que a quantidade de "[ ]" indica de quantas dimensões será a matriz, caso houver apenas um "[ ]" então é um vetor.

### Classes

Por ser uma linguagem totalmente orientada a objetos não é possível criar estruturas, apenas Classes, que pode ser feito da seguinte forma:

```
class <nomeDaClasse> {  
  
    <atributos>  
    <métodos>  
  
}
```

## Instruções condicionais

A declaração de instruções condicionais deve ser da forma:

```
if (<condição>){  
  
    //código  
  
}
```

Ou:

```
if (<condição>){  
  
    //código  
  
}else{  
  
    //código  
  
}
```

Ou ainda:

```
<variavel> = <condicao>?<valor-se-true>:<valor-se-false>;
```

Exemplo:

```
int a = 2>3? 2:3;
```

Se 2 for maior que 3 então a recebe o valor 2, senão recebe o valor 3. Neste exemplo, "a" vai receber o valor 3.

## Instruções de repetição

As instruções de repetição são:

## for

### Declaração:

```
for (<inicializacao de variaveis>;<condicao>;<incremento>){  
  
    //código  
  
}
```

### Exemplo:

```
for (int i = 0; i < 10; i++){  
  
    System.out.println(i);  
  
}
```

## for-each

### Declaração:

```
for(<tipo-objeto> <identificação> : <coleção>){  
  
    //código  
  
}
```

### Exemplo:

```
int[] meses = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };  
for (int m : meses) {  
  
    System.out.println(m);  
  
}
```

## while

### Declaração:

```
while(<condição>){  
  
    //código  
  
}
```

## do while

### Declaração:

```
do{
```

```
    //código  
}while(<condição>);
```

## Métodos

A declaração de métodos deve ser feita da seguinte forma:

```
<tipo-de-retorno> <nome>(<parâmetros>){  
    //código  
}
```

Exemplo:

```
int dobro(int numero){  
    return numero * 2;  
}
```

## Objetos

Objetos podem ser criados invocando um construtor da classe:

```
<tipo-do-objeto> <nome> = new <construtor-do-objeto>
```

Por exemplo:

```
File f = new File("c:\f1.txt");
```

## Comentários

Comentários podem ser de dois tipos:

Uma linha:

```
//comentário
```

Várias linhas:

```
/* inicio de comentário
```

```
ainda comentário  
fim de comentário */
```

## Exceções

Para realizar controle de exceções deve-se seguir a seguinte forma:

```
try{  
    //código passível de exceções  
}catch( <exceção> e){  
    //capturando a exceção e  
}finally{  
    //sempre executa esse código, ocorrendo ou não a  
    //exceção  
}
```

Sendo que *finally* ou *catch* devem estar presentes quando houver um bloco *try*. É permitido, mas não é necessário que os dois estejam presentes ao mesmo tempo.

Java fornece uma vasta API para manuseio de arquivos. Um exemplo de como utilizá-la segue abaixo:

```
import java.io.*;  
class EscreverEmArquivo {  
    public static void main (String args[]) {  
        FileOutputStream fout;  
        try {  
            fout = new FileOutputStream ("arquivo.txt");  
            // imprime uma linha no arquivo  
            new PrintStream(fout).println ("hello world!");  
            fout.close();  
        } catch (IOException e){
```

```

        System.err.println ("Não foi possível escrever no
arquivo");
        System.exit(-1);
    }
}
}

class LerDeArquivo
{
    public static void main (String args[]) {
        FileInputStream fin;
        try {
            fin = new FileInputStream ("arquivo.txt");
            //Lê uma linha de texto
            System.out.println(new
DataInputStream(fin).readLine());
            fin.close();
        }catch (IOException e){
            System.err.println ("Não foi possível ler
arquivo");
            System.exit(-1);
        }
    }
}
}

```

## 6. Tipo de aplicações

O Java pode ser utilizado para o desenvolvimento de aplicações Desktop, Web e Móvel.

## 7. IDE disponíveis

- JEdit - Software livre, recomendado para programadores iniciantes. [<http://www.jedit.org>]
- JCreator - Comercial, valor entre 89 e 5.000 dólares recomendado para programadores iniciantes. [[www.jcreator.com](http://www.jcreator.com)]

- BlueJ - Software livre, recomendado para programadores iniciantes. [http://www.bluej.org]
- Eclipse - Software livre, é um projeto aberto iniciado pela IBM. [http://www.eclipse.org]
- NetBeans - Software livre, ambiente criado pela empresa Sun Microsystems. [www.netbeans.org]
- JBuilder - Comercial, valor entre 465 e 1.495 dólares, ambiente criado pela empresa Borland. [www.codegear.com/products/jbuilder]
- JDeveloper - Gratuito, IDE desenvolvida pela empresa Oracle. [www.oracle.com/technology/products/jdev]
- IntelliJ IDEA - Comercial, valor entre 250 e 600 dólares, IDE desenvolvida pela JetBrains. [www.jetbrains.com/idea/]

## 8. Padronização ISO/ANSI

A linguagem JAVA ainda não foi padronizada pela ISO/IEC JTC 1.

## 9. Conexão com Banco de Dados

A conexão com banco de dados em Java é feita utilizando a API JDBC (Java Data Base Connectivity). A JDBC define interfaces e classes para escrever aplicações com utilização de banco de dados.

Segue abaixo um exemplo simples de conexão com banco de dados e exibição do resultado parcial de um select:

```
Class.forName("<classe do driver jdbc>");

Connection con =
DriverManager.getConnection("<urljdbc>", "<usuario>", "<senha
>");

Statement stm = con.createStatement();

ResultSet rs = stm.executeQuery("select * from tabela");

while(rs.next()) {
    System.out.println("coluna 1:" + rs.getString(1));
}

con.close();
```



## 10. Interação com outras linguagens

A interação é feita através da Java Native Interface (JNI). Usando a JNI na linguagem Java é possível ter uma comunicação com bibliotecas feitas em outras linguagens como C, C++.

Esta interação pode ser feita tanto no Windows (DLL) quanto no Linux (SO).

Segue abaixo um simples código demonstrando a utilização da JNI.

```
class firstJNI
{
    public native void showHelloWorld();
    private native String getAbout();

    static {
        System.loadLibrary("firstjni"); // firstjni.dll
    }

    public static void main(String[] args)
    {
        firstJNI JN = new firstJNI();
        JN.showHelloWorld();
        String about = JN.getAbout();
        System.out.println("About:" + about);
    }
}
```

## 11. Javadoc

Javadoc é uma ferramenta da SUN para geração da documentação do código com base nos comentários. A documentação é gerada no formato html. Segue abaixo um exemplo de comentário para documentação de uma função [6]:

```
/**
 * Returns an Image object that can then be painted on the screen.
 * The url argument must specify an absolute {@link URL}. The name
 * argument is a specifier that is relative to the url argument.
 * <p>
 * This method always returns immediately, whether or not the
 * image exists. When this applet attempts to draw the image on
 * the screen, the data will be loaded. The graphics primitives
 * that draw the image will incrementally paint on the screen.
 *
 * @param url an absolute URL giving the base location of the image
 * @param name the location of the image, relative to the url argument
 * @return the image at the specified URL
```

```

* @see Image
*/
public Image getImage(URL url, String name) {
    try {
        return getImage(new URL(url, name));
    } catch (MalformedURLException e) {
        return null;
    }
}

```

O resultado da documentação acima seria:

### **getImage**

```
public Image getImage(URL url,String name)
```

Returns an Image object that can then be painted on the screen. The url argument must specify an absolute [URL](#). The name argument is a specifier that is relative to the url argument.

This method always returns immediately, whether or not the image exists. When this applet attempts to draw the image on the screen, the data will be loaded. The graphics primitives that draw the image will incrementally paint on the screen.

#### **Parameters:**

url - an absolute URL giving the base location of the  
 imagename - the location of the image, relative to the url argument

**Returns:** the image at the specified URL

**See Also:** [Image](#)

## **12. RichFaces**

RichFaces é uma biblioteca de componentes para aplicações Web que utilizam o framework JSF. Com ele é possível utilizar componentes como calendários, listas, menus, etc, de forma muito simples. O principal objetivo do framework é tornar aplicativos Web parecidos com os aplicativos desktop [7].

Os componentes do RichFaces ajudam no ganho de produtividade por fazer uso das funcionalidades dos componentes integrados para criar aplicações Web que proporcionam melhoria da experiência do usuário sendo muito mais confiável e mais rápido. O RichFaces inclui também um forte apoio para o JSF também tira o máximo dos benefícios da JSF incluindo ciclo de vida, a

validação, conversão e instalações, juntamente com a gestão de recursos estáticos e dinâmicos. [8]

## **12.1. Características**

Compatível com as características do JSF.

- Validação e conversão
- Ciclo de vida JSF.

Adiciona comportamento Ajax em aplicações já existentes.

- Page-Wide Ajax Support

Suporte a Skins.

## **12.2. Licença**

O RichFaces está sob a licença GNU Lesser General Public License (LGPL), que é considerada um meio termo entre a GPL e licenças mais permissivas, tais como a licença BSD e a licença MIT. Estar sob a licença LGPL significa que qualquer um pode:

- Copiar, distribuir, exibir e executar o software;
- Criar outro software baseado nele e fazer uso comercial do software.

Desde que sejam observadas as seguintes condições:

- Deve dar crédito ao autor original;
- Para cada novo uso ou distribuição, deve deixar claro para terceiros os termos da licença deste software;
- Qualquer uma destas condições pode ser renunciada, desde que seja obtida a permissão do autor;
- O software é distribuído no estado em que se encontra, sem custos, e sem qualquer garantia, implícita ou explícita, por parte do autor.

### 12.3. Sistemas Suportados

Por ser implementado totalmente em Java, o framework pode ser utilizado nas plataformas X86, x64, Intel Itanium, SPARC, Java Platform Standard Edition e nos Sistemas Operacionais Windows, Linux, Solaris e Apple OS X.

### 12.4. Executar um exemplo simples no Jboss 5.x

Para compilar um exemplo simples utilizando o Richfaces no Jboss 5.x é necessário os seguintes procedimentos, considerando uma aplicação web simples já funcionando:

1. Adicionar as bibliotecas do Richfaces no projeto contidas no \*.bin.zip ou \*.bin.tar.gz em <http://labs.jboss.com/jbossrichfaces/downloads>.
2. Registrar o JSF1.2 e o Richfaces no web.xml como a seguir:

```
<filter>
<display-name>RichFaces Filter</display-name>
<filter-name>richfaces</filter-name>
<filter-class>org.ajax4jsf.Filter</filter-class>
</filter>

<filter-mapping>
<filter-name>richfaces</filter-name>
<servlet-name>Faces Servlet</servlet-name>
<dispatcher>REQUEST</dispatcher>
<dispatcher>FORWARD</dispatcher>
<dispatcher>INCLUDE</dispatcher>
</filter-mapping>

<listener>
<listener-
class>com.sun.faces.config.ConfigureListener</listener-
class>
</listener>

<!-- Faces Servlet -->
<servlet>
<servlet-name>Faces Servlet</servlet-name>
<servlet-class>javax.faces.webapp.FacesServlet</servlet-
class>
<load-on-startup>1</load-on-startup>
</servlet>

<!-- Faces Servlet Mapping -->
<servlet-mapping>
```

```
<servlet-name>Faces Servlet</servlet-name>  
<url-pattern>*.jsf</url-pattern>  
</servlet-mapping>
```

### 3. Adicionar as taglib nos arquivos:

#### Declaração em páginas JSP

```
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>  
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>  
<!-- RichFaces tag library declaration -->  
<%@ taglib uri="http://richfaces.org/a4j" prefix="a4j"%>  
<%@ taglib uri="http://richfaces.org/rich" prefix="rich"%>
```

#### Para paginas XHTML

```
<xmlns:a4j="http://richfaces.org/a4j">  
<xmlns:rich="http://richfaces.org/rich">
```

## Referências

[1] Wikipedia. Disponível em [http://pt.wikipedia.org/wiki/Java\\_\(linguagem\\_de\\_programa%C3%A7%C3%A3o\)](http://pt.wikipedia.org/wiki/Java_(linguagem_de_programa%C3%A7%C3%A3o)). Acessado em 28 de abril de 2009.

[2] Companhia de Docas do Rio Grande do norte (CODERN). Concurso público para provimento de vagas para o cargo de Programador. Prova disponível em <http://www.comperve.ufrn.br/conteudo/concursos/codern/provas/204.pdf>. Acessado em 28 de abril de 2009.

[3] TRIBUNAL REGIONAL ELEITORAL DE MINAS GERAIS. Concurso Público para provimento de cargos de Judiciário - Área Apoio Especializado. Especialidade Programação de Sistemas. Prova disponível em [http://www.questoesdeconcursos.com.br/prova/arquivo\\_prova/297/FCC\\_2007\\_TRE-MG\\_Tecnico-Judiciario-Programacao-Sistemas.pdf](http://www.questoesdeconcursos.com.br/prova/arquivo_prova/297/FCC_2007_TRE-MG_Tecnico-Judiciario-Programacao-Sistemas.pdf). Acessado em 28 de abril de 2009.

[4] Download JVM Java. Disponível em <http://java.com/en/download/index.jsp>. Acessado em 28 de abril de 2009.

[5] Download JDK. Disponível em <http://java.sun.com/javase/downloads/index.jsp>. Acessado em 28 de abril de 2009.

[6] How to Write Doc Comments for the Javadoc Tool. Disponível em <http://java.sun.com/j2se/javadoc/writingdoccomments/index.html>. Acessado em 28 de abril de 2009.

[7] An introduction to RichFaces. Disponível em <http://www.ibm.com/developerworks/web/library/j-richfaces/index.html>. Acessado em 28 de abril de 2009.

[8] JBoss Community. Disponível em <http://www.jboss.org/jbossrichfaces>. Acessado em 28 de abril de 2009.

[9] Wikipedia. Disponível em [http://en.wikipedia.org/wiki/Java\\_version\\_history](http://en.wikipedia.org/wiki/Java_version_history). Acessado em 28 de abril de 2009.