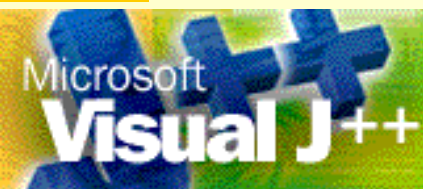


desenvolvimento de aplicações

WEB

com

JAVA



Helder Rocha (hlsr@uol.com.br)

```
for (x = 0; x < 1  
repeat !"less"  
if (language  
document.w
```



Objetivos

1

Apresentar a Linguagem Java

- características, potencial, aplicações
- últimas novidades

2

Exemplos de desenvolvimento Java

- uma applet simples
- uso de JDBC para acesso a BDs relacionais

3

Ambientes de desenvolvimento

- principais produtos



```
for (x = 0; x < 1  
repeat {"less"  
if (language  
document.w
```



Webs e Intranets com Java

- A tecnologia Java é uma ótima solução para o desenvolvimento de aplicações distribuídas baseadas na arquitetura de Web. Vantagens:
 - interface uniforme
 - funcionalidade superior à obtida com HTML + JavaScript + CGI
 - independência de plataforma
- Formas de implementação na Web
 - **Applets:** componentes simples (arquivos .class)
 - **JARs: Applets + JavaBeans:** componentes reutilizáveis, comprimidos, autenticáveis (arquivos .jar)
- Comunicação e integração com ActiveX/DCOM



```
for (x = 0; x < 1  
repeat ("less"  
if (language  
document.w
```



Aplicações Java

- Há dois tipos básicos de aplicações Java:
 - **componentes (applets/JARs)** : aplicações que executam dentro de um browser
 - **aplicações *standalone*** : aplicações que rodam independentes de browser (como qualquer outra aplicação do desktop)
- Aplicações Java rodam em qualquer lugar
 - dependem apenas da existência de um **emulador de máquina Java** no sistema onde rodam (os browsers mais modernos têm emuladores embutidos)



```
for (x = 0; x < 1  
repeat {"less  
if (language  
document.w
```



Applets

- Não confunda “Java”, a linguagem, com “Applets”, os componentes:

- Applets são **programas** compilados para linguagem de máquina de um processador Java (*.class).
- Applets rodam no browser, que interpreta o *código de máquina* via emulador (Java Virtual Machine)

- Usa-se HTML para incluir applet em área de uma página e passar parâmetros:

```
<applet code="URL da applet" height=50 width=50>  
  <param name="variavel1" value="valor">  
  <param name="variavel2" value="valor">  
</applet>
```

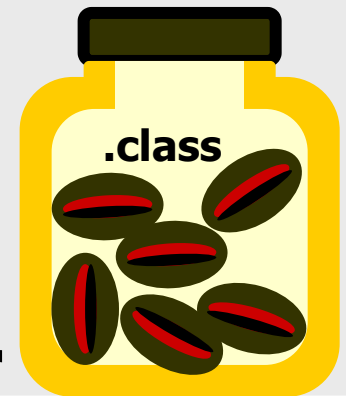


```
for (x = 0; x < 1  
repeat !*less  
if (language  
document.w
```



Java Beans

- Arquitetura de componentes implementada na API Java
 - propriedades, introspecção, eventos plugáveis
 - controle de versões
 - certificados de segurança
 - comunicação entre applets e outros componentes
- Para criar um bean, basta seguir algumas convenções
- Na Web, beans devem ser distribuídos em JARs
 - Formato tipo ZIP, com meta-informação
 - Meta-informação do bean define versões, segurança, etc.



.jar



```
for (x = 0; x < 1  
repeat ("less:  
if (language  
document.w
```



O que NÃO é Java

- JavaScript **NÃO É** a mesma coisa que Java!
- JavaScript é uma linguagem de roteiro (script), de propriedade da Netscape
- Código-fonte JavaScript é embutido no HTML e interpretado linha-por-linha pelo browser.
 - Não se inclui Java em uma página, se carrega uma applet ou componente
- JavaScript estende as capacidades limitadas do HTML
 - manipula elementos HTML como objetos
 - interage com applets (JavaScript 1.1 e 1.2)



```
for (x = 0; x < 1  
repeat {"less"  
if (language  
document.w
```



O que NÃO é Java

- Java **não é** comparável a ActiveX
- ActiveX sequer é uma linguagem...
 - Componentes ActiveX podem ser implementados em C++, Visual Basic e até Java!
- ActiveX é uma arquitetura de componentes (é a mesma coisa que OLE!)
 - Não faz sentido comparar ActiveX com **linguagem** Java
 - Faz sentido comparar ActiveX com a **tecnologia** Java (applets, Java Beans)
- É um padrão aberto: <http://www.activex.org>
- Suportado pelo MSIE 3.x, 4.x e Netscape 4.x




```
for (x = 0; x < 1  
repeat ("less  
if (language  
document.w
```



Aplicações “standalone”

- Java pode ser usada para desenvolver aplicações independentes de browser.
- Vantagens: Distribuição de software
 - distribuição de único pacote para instalação/execução em múltiplas plataformas
 - instalação traduz código Java para código nativo
 - acesso a bibliotecas nativas via métodos ou diretamente (API Win32s através do J/Direct)
- Interpretação eficiente de código de máquina Java com JIT compiler agiliza desenvolvimento



```
for (x = 0; x < 1  
repeat {"less"  
if (language  
document.w
```



O que muda com Java

- Conseqüências da filosofia de Java na engenharia de software:
 - maior importância da aplicação sobre o SO
 - formas mais eficientes de distribuição de software
 - atualização dinâmica simplificada
 - desenvolvimento multiplataforma
 - pode-se desenvolver em Sun para rodar em Macintosh, Windows, etc.
- Java já nasceu adaptada a padrões emergentes
 - suporte a Unicode (tipo char de 16 bits)
 - tratamento nativo de URLs

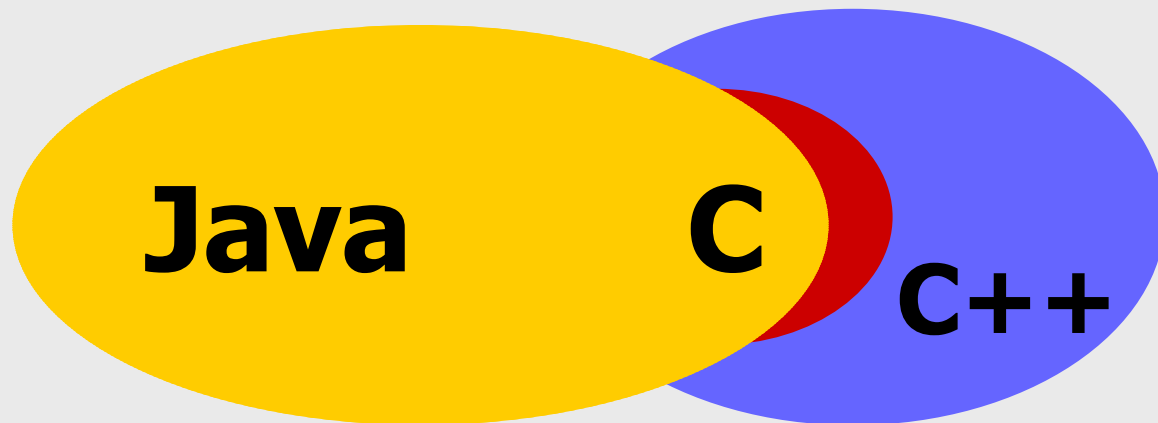


```
for (x = 0; x < 1  
repeat (*less  
if (language  
document.w
```



A linguagem Java

- Java não é somente outra sigla qualquer (**J**ust **A**nother **V**ague **A**cronym)
- Java é uma **linguagem de programação** com sintaxe semelhante a de C, porém orientada a objetos
 - não é híbrida como C++: é **impossível** escrever um programa em Java sem usar objetos



```
for (x = 0; x < 1  
repeat ("less"  
if (language  
document.w
```



Por que Java?

- Java é popular. Todo mundo está usando ou vai usar
- Fatores que contribuem para a popularidade e aceitação de Java
 - semelhança com C e C++
 - Java é mais simples que C e tão poderoso quanto C++
 - aplicação na Internet e Intranet
 - 100% portátil
 - binários independentes de plataforma



```
for (x = 0; x < 1  
repeat ("less"  
if (language  
document.w
```



Java comparada a C/C++

- Java é bem **mais simples** que C/C++
 - Não tem ponteiros (não tem acesso direto a posições de memória)
 - Não tem #defines, #includes, arquivos ".h", protótipos
- Java é **mais robusta**
 - Tem um sistema de coleta de lixo, que dispensa liberação de memória (não tem delete)
 - Usa exceções para controlar erros (como Delphi)
- É uma evolução do C para orientação a objetos, sem a complexidade de C++



```
for (x = 0; x < 1  
repeat!"less:  
if (language  
document.w
```



Java parece com C

- Exemplo de sintaxe (definição de método):

```
/* método somatório */  
int soma (int vetor[]) {  
    int resultado = 0;  
    for (int i=vetor.length; --i >= 0; )  
    {  
        resultado += vetor[i];  
    }  
    return resultado;  
}
```



```
for (x = 0; x < 1  
repeat ("less"  
if (language  
document.w
```



Robustez e Segurança

- Rigorosa verificação dos **tipos**
- Controle de **exceções** é obrigatório
- **Coleta de lixo** automática dispensa liberação de memória
- Compilador verifica limites de vetores e referências vazias
 - não provoca GPFs ou core-dumps
- Bytecode (linguagem de máquina Java) contém informações que permitem a verificação da integridade do código
 - segurança

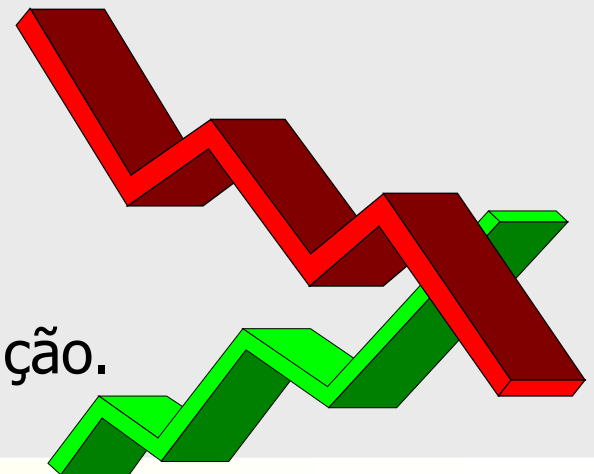


```
for (x = 0; x < 10; x++)  
  repeat { *less:  
    if (language:  
      document.*
```



Desempenho

- Interpretação de bytecodes pela JVM: **10 a 30 vezes mais lento** que C/C++
 - Para maioria das **aplicações GUI** esse valor é aceitável
 - Inaceitável para aplicações ou rotinas que exigem muito processamento
- Máquina Virtual com compilação **Just-in-Time (JIT)** consegue uma eficiência **20 vezes maior**
 - Os principais browsers possuem JITs instalados
 - Compilação ocorre antes da execução.




```
for (x = 0; x < 1  
repeat: *less:  
if (language:  
document.w
```



Restrições de applets remotos

- As restrições sobre applets remotos limitam o desenvolvimento de applets “úteis” em Java
- Sem permissão para escrever no disco do cliente, ficam inviáveis aplicações do NC.
- Não podem escrever em disco, não podem imprimir... para que servem?
- Solução: **JARs (Java 1.1)**
 - contém informações sobre segurança (certificados)
 - applets remotos podem ser considerados “confiáveis” e terem a mesmas regalias que applets locais
- Mesma solução é possível com ActiveX



```
for (x = 0; x < 1  
repeat {"less"  
if (language  
document.w
```



Multithreading

- Múltiplas linhas de execução definidas a nível de programação
- Controle de prioridades e de sincronização
 - métodos sincronizados nunca são chamados ao mesmo tempo
 - garante integridade dos dados
 - risco de deadlock
- Maior desempenho em sistemas multiprocessados



```
for (x = 0; x < 1  
repeat {"less"  
if (language  
document.w
```



Java e POO

- Programação orientada a objetos
 - realizada através da interação entre objetos
 - centralizada na **classe** (representa um **objeto** com **estado** e **comportamento**)
- Programação orientada a procedimentos
 - realizada definindo passos de um procedimento
 - centralizada no procedimento (representa uma série de instruções)



```
for (x = 0; x < 1  
repeat {"less"  
if (language  
document.w
```



POO: Conceitos

- Todos os dados manipulados em um programa em Java, exceto variáveis de tipos básicos, são objetos
- As características básicas de linguagens orientadas a objetos são
 - abstração
 - herança e polimorfismo
 - encapsulamento
- Objetos são **abstrações** de modelos reais (ou imaginários) que tem estado e comportamento
 - O **estado**, é descrito por campos de dados, que podem ser variáveis
 - O **comportamento** é descrito por métodos (funções)



```
for (x = 0; x < 1  
repeat {"less"  
if (language  
document.w
```



POO: Conceitos

- O **encapsulamento** permite que se esconda a complexidade do código por trás dos métodos
 - objetos ficam fáceis de usar (toda a funcionalidade está nos métodos)
 - dados são protegidos (só são modificados por métodos)
- A **herança** permite que se construa em cima de uma funcionalidade já existente
 - simplicidade maior ainda: para fazer um pneu, não é preciso reinventar a roda
 - Classe herda implementação de outra. Métodos mantêm o mesmo nome (**polimorfismo**)





Programa mínimo em Java

- A palavra-chave "class" define uma classe em Java.
- O menor programa em Java é uma classe:

```
class Programinha {  
}
```

- Salve-o como: *prog.java*
 - Pode ter qualquer nome desde que preserve a extensão **.java**
- Ele não faz nada, porém irá compilar sem erros:
 - C:\> **javac** prog.javae gerar um arquivo binário **Programinha.class**



```
for (x = 0; x < 1  
repeat "less"  
if (language  
document.w
```



O programa é um objeto

- A “classe” representa o programa-objeto em tempo de execução
 - Há **uma** classe
 - Pode haver **vários** objetos
- O objeto é uma **instância** de uma classe
- Pode-se criar novos objetos a partir de classes existentes:

```
Programinha pg = new Programinha();  
String s = new String("Eu sou uma String");
```





Tipos, variáveis, referências

- O **estado** de um objeto é representado por variáveis ou referências
- Referências são objetos. Variáveis são tipos básicos como inteiros, caracteres, etc.

```
public class Programinha {  
    public int x = 10;  
    private float y = 1.41;  
    String s = new String("Eu sou uma String");  
}
```

- Tipos tem que ser **declarados** antes de usados
- "new" cria um objeto a partir de um construtor
- "public" e "private" controlam acesso




```
for (x = 0; x < 1  
repeat {"less"  
if (language  
document.w
```



Tipos de dados básicos

- **boolean**
 - (8 bits; 1 de informação)
- **byte**
 - (8 bits, com sinal)
- **char**
 - (16 bits, sem sinal, Unicode)
- **short**
 - (16 bits, com sinal)
- **int**
 - (32 bits, com sinal)
- **long**
 - (64 bits, com sinal)
- **float**
 - (32 bits, IEEE 754)
- **double**
 - (64 bits, IEEE 754)

Tipos são iguais em todas as plataformas





```
for (x = 0; x < 1  
repeat ("less  
if (language  
document.w
```

Tipos complexos

- Classes definem tipos mais complexos
 - Usadas para criar objetos
- Em Java, uma cadeia de caracteres é um objeto, definido pela classe **String**:

```
String s;
```

```
s = new String("Sou uma String!");
```

ou

```
s = "Sou uma String!"; // sintaxe só válida  
                        // para a classe String
```





Métodos

- Métodos são meio de comunicação com outros objetos

- são os controles que alteram os dados
- Sintaxe:

[modificador] tipo_retorno **método**(parametros) {...}

- Método square()

```
class Programinha {  
    public int x = 10;  
  
    public int square() {  
        return x * x;  
    }  
}
```





```
for (x = 0; x < 1  
repeat!"less"  
if (language  
document.w
```

Objetos

■ Comunicação entre objetos

```
public class Programão {  
    Programinha pg = new Programinha();  
    int q = pg.square;    // q contém 100  
    int p = pg.x;        // p contém 10  
    int r = pg.y;        // erro: pg.y é private  
}
```

- Métodos e variáveis são acessíveis via objetos
- Somente métodos e variáveis públicas são acessíveis de fora do objeto
- // são comentários. /* ... */ também
- Nota: se a classe for "public", nome do arquivo *.java tem que ter o mesmo nome que a classe



```
for (x = 0; x < 1  
repeat "less"  
if (language  
document.w
```



Encapsulamento

```
class Programinha {  
    private int x = 10;  
  
    public int getX() {  
        return x;  
    }  
  
    public void setX(int newX) {  
        x = newX;  
    }  
}
```

- Só é possível ler ou alterar X através de métodos:
 - Programinha pg = new Programinha();
pg.setX(40);
int leiaX = pg.getX();



```
for (x = 0; x < 1  
repeat: *less:  
if (language:  
document.w
```



Herança

- Pode-se construir sobre infraestrutura existente
- O programa abaixo herda toda a estrutura de uma applet:

```
public class Programinha extends java.applet.Applet {  
    ...  
}
```

ou

```
import java.applet.Applet;  
public class Programinha extends Applet {  
    ...  
}
```

- "import" indica o caminho da(s) classe(s) usada(s)

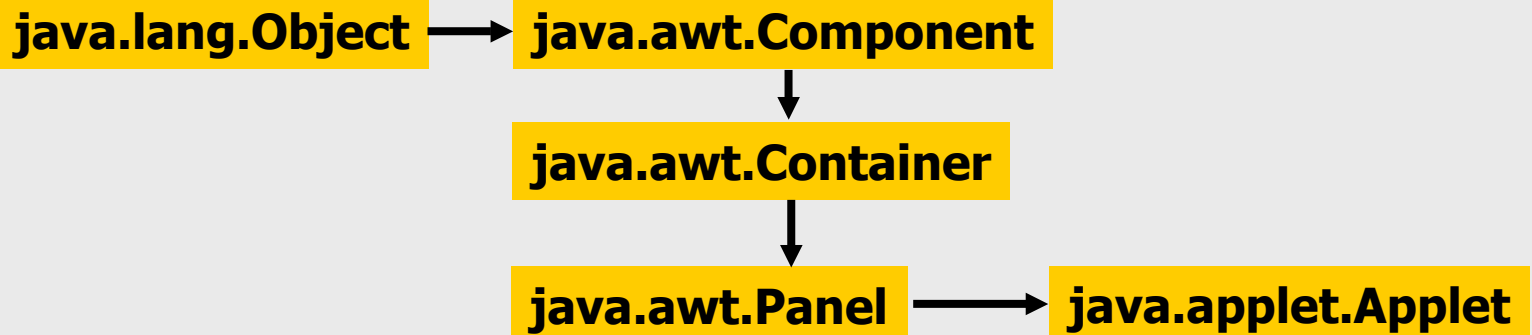


```
for (x = 0; x < 1  
repeat ("less  
if (language  
document.w
```



Herança

- Por default, toda classe estende **Object**.
- Para estender outra classe, usa-se a palavra-chave "extends"
- Hierarquia da classe `java.applet.Applet`



```
for (x = 0; x < 1  
repeat!"less:  
if (language  
document.w
```



Uma Applet finalmente...

```
import java.awt.*;  
import java.applet.Applet;  
public class Programinha extends Applet {  
    private int x = 10;  
    String s = new String("Eu sou uma Applet!");  
  
    public void paint(Graphics g) {  
        g.drawString(s, x*2, 50);  
    }  
}
```

- Salve como "Programinha.java"
- Compile: `javac Programinha.java`




```
for (x = 0; x < 1  
repeat!"less:  
if (language:  
document.w
```



Inclusão na página web

- É necessário saber a URL da applet e o tamanho ocupado na tela
- Coloque o arquivo **Programinha.class** (gerado na compilação) no mesmo diretório que o arquivo HTML a seguir:

- ```
<html>
 <applet code="Programinha.class"
 height=100
 width=300>
 </applet>
</html>
```

**Eu sou uma Applet!**

- Carregue o arquivo HTML no browser... e pronto!



```
for (x = 0; x < 1
repeat ("less:
if (language:
document.w
```



# Interface Gráfica

- O “pacote” `java.awt` contém as ferramentas básicas da interface gráfica como botões, scrollbars, etc.
- Para criar uma interface gráfica na applet é preciso
  - 1. importar a classe do `java.awt` (ou importar toda a `java.awt`)
  - 2. criar o objeto instanciando sua classe (botão, combo-box, etc.)
  - 3. colocar o objeto no applet (método `add`)
  - 4. inicializar a applet com os componentes (método `init`)
  - 5. tratar os eventos relacionados (sobrepondo métodos herdados da classe `Component`)



```
for (x = 0; x < 1
repeat!"less:
if (language:
document.w
```



# Exemplo (interface)

```
import java.awt.*; // (1) importação de todas as classes java.awt
import java.applet.Applet;
public class Programinha extends Applet {
 Button b1; // (2) declaração de referências
 TextField t1;

 public void init() { // (4) método de inicialização
 t1 = new TextField(20); // (2) criação de objetos
 b1 = new Button("Me aperte!");
 add(b1); // (3) adição de objetos na applet
 add(t1);
 }
}
```

Me aperte!



```
for (x = 0; x < 1
repeat!*less
if (language
document.w
```



## Exemplo (interface + eventos)

(...)

```
public void init() { // (4) método de inicialização
 t1 = new TextField(20); // (2) criação de objetos
 b1 = new Button("Me aperte!");
 add(b1); // (3) adição de objetos na applet
 add(t1);
}
```

```
public boolean action (Event evt, Object oque) { // (5) evento
 if (evt.target == b1) {
 t1.setText("Você me apertou!");
 return true;
 } else return false;
}
}
```

Me aperte!

Você me apertou!





# Alternativas de conectividade a BDs

- Bancos de Dados Relacionais
  - JDBC (Java Database Connectivity)
- Bancos de Dados Orientados a Objetos
  - ODMG, capítulo 7
- Integração BD Relacional-BD Objeto
- Serialização de objetos: persistência
  - API RMI/Serialization



```
for (x = 0; x < 1
repeat (*less
if (language
document.w
```



# O que é JDBC?

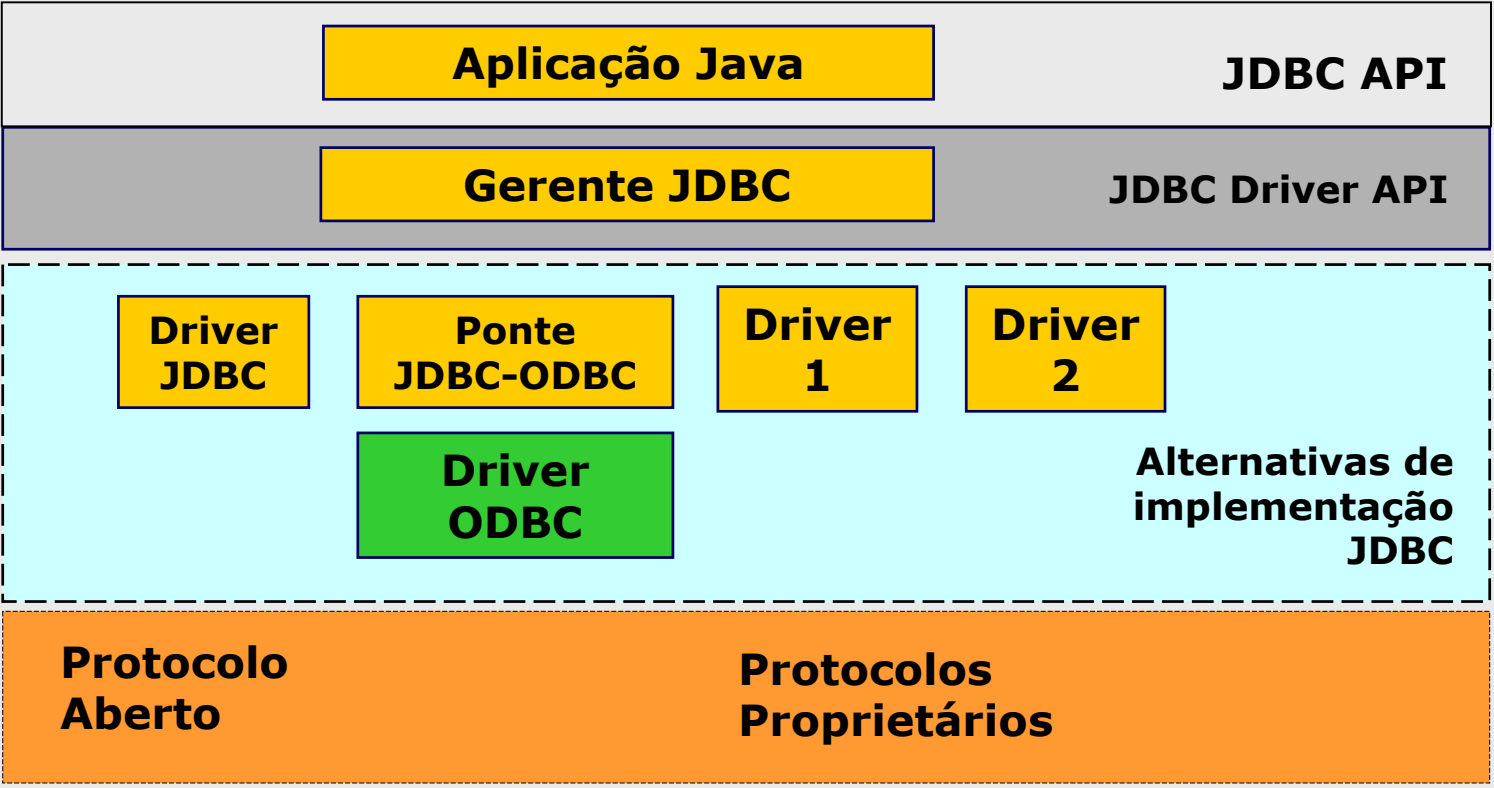
- **Java Database Connectivity:** uma interface padrão de acesso a bancos de dados SQL oferecendo acesso uniforme a qualquer BD.
  - Pode-se escrever um único programa que será capaz de enviar instruções SQL para o banco de dados correto, utilizando um driver adequado.
- É uma API independente de SGBD
- Define classes Java para representar
  - conexões de rede, declarações SQL, conjuntos de resultados, conjuntos de dados
- Baseado no padrão X/Open Call Level Interface (CLI)
  - Espelhado no ODBC





```
for (x = 0; x < 1
repeat (*less
if (language
document.*
```

# Diagrama de Camadas JDBC



```
for (x = 0; x < 1
repeat ("less"
if (language
document.w
```



# O que JDBC faz?

- Em resumo, JDBC dá condições de realizar três operações:
  - estabelecer uma conexão com um banco de dados
  - enviar instruções SQL
  - recuperar e processar os resultados.
- JDBC usa "interfaces" Java
  - Interfaces são classes especiais. Uma classe pode implementar várias interfaces, herdando os protótipos de seus métodos.
  - Métodos declarados em "interfaces" não têm implementação. A implementação está nas classes que implementam as interfaces (os drivers JDBC).







```
repeat!*less:
if (language:
document.*
```

# Exemplo: Acesso a BDs

- O Banco de Dados "biblio" contém a seguinte tabela "Livro"

## Livro

Título	Autor	Editora	ISBN
Sphere	Michael Crichton	Ballantine	0-345-35314-5
Mr. Vertigo	Paul Auster	Faber and Faber	0-571-17345-4
The Lost World	Michael Crichton	Ballantine	0-345-40507-2
<b>A Nova Califórnia</b>	Lima Barreto	Círculo do Livro	85-332-0524-4
Cronica de una Muerte Anunciada	Gabriel Garcia Marquez	Plaza & Janes	84-01-42270-01
Interview with the Vampire	Anne Rice	Ballantine	0-345-33766-2
Livro de Sonetos	Vinícius de Moraes	Círculo do Livro	85-332-0139-5

- O slide a seguir mostra um exemplo de conexão JDBC



```
for (x = 0; x < 1
repeat "less
if (language
document.w
```



# JDBC: Exemplo

```
Connection con = DriverManager.getConnection("jdbc:odbc:biblio",
 "login",
 "passwd");
```

```
Statement stmt =
 con.createStatement();
```

```
ResultSet rs =
 stmt.executeQuery(" SELECT Título, ISBN
 FROM Livro
 WHERE Autor = 'Lima Barreto'");
```

```
while (rs.next()) {
 int x = rs.getInt("ISBN");
 String s = rs.getString("Autor");
 float f = rs.getFloat("Preço");
}
```



```
for (x = 0; x < 1
repeat (*less
if (language
document.w
```



# Aplicações

- Applets podem usar JDBC para acesso a Bancos de Dados via Internet ou na Intranet
- Restrições de segurança dos Applets podem ser flexibilizadas através de certificados (usando JARs) para ter mais acesso.
- Aplicações C/S independentes de browser rodando em uma Intranet podem acessar diferentes SGBDs em diferentes plataformas
- Uma única interface pode ser usada para vários sistemas diferentes



```
for (x = 0; x < 1
repeat {"less
if (language
document.w
```



# A Internet, Java e o Futuro

- Últimos avanços na área incluem:
  - Sun Microsystems e Netscape:
    - **Java 1.1:** JDBC, RMI (Remote Method Invocation), Java Beans, JARs, certificados de segurança, internacionalização, novo modelo de eventos da GUI.
  - Microsoft:
    - **ActiveX** e tecnologias relacionadas
    - **J/Direct:** arquitetura que permite acesso direto à API Win32s
- No futuro, Java pode substituir C++ como linguagem preferencial para aplicações no desktop
  - J/Direct e tecnologias similares podem antecipar futuro



```
for (x = 0; x < 1
repeat {"less:
if (language:
document.w
```



# Principais novidades do JDK 1.1

- **Desempenho** do JVM: 2 a 5x mais rápido + JIT
- **AWT**: totalmente redesenhado; novo modelo de eventos
- **JavaBeans**
- Suporte a **assinaturas digitais** e outros recursos de segurança
- **Java Archive Format**: ZIP com metainformação. Única conexão HTTP para descarregar applets + arquivos
- **Remote Method Invocation (RMI)**
- **Internacionalização**
- **JDBC**
- **Java Native Interface**: compatibilidade em mesma plataforma
- **Reflexão**: permite descoberta de informação de classes carregadas dinamicamente
- **Serialização de objetos**



```
for (x = 0; x < 1
repeat {"less
if (language
document.w
```



# Estado atual da linguagem

- Versão 1.1.2 (mais popular ainda é 1.02)
- Distribuição **gratuita**, através da Internet, do JDK (Java Development Kit)
- Plataformas suportadas pela Sun
  - Windows95, NT, Solaris 2.3, Mac 7.5
- Plataformas suportadas por outras empresas
  - Linux, AT&T Unix, HP/UX, OS/2, SGI, W31
- Java não roda de forma consistente em plataformas de 16 bits que não suportam multithreading e nomes longos de arquivos (Win3.11)



```
for (x = 0; x < 1
repeat {"less"
if (language
document.w
```

# Ambientes Integrados de Desenvolvimento

## ■ SunSoft (<http://www.sun.com/sunsoft>)

- Java WorkShop (\$99)
- Java Studio (preview)
  - SPARC, W95/NT



## ■ Microsoft (<http://www.microsoft.com/products>)

- Visual J++ (\$99)



## ■ Aimtech (<http://www.jamba.com>)

- (\$295)



```
for (x = 0; x < 1
repeat: *less:
if (language:
document.w
```

# Ambientes Integrados de Desenvolvimento

- Penumbra Software  
(<http://www.penumbrosoft.com>)
  - Mojo Enterprise(\$195/\$895)
- Symantec (<http://cafe.symantec.com>)
  - Café (\$79.95)
  - Visual Café Pro (\$499.95)
    - ambiente visual com geração de interface gráfica e eventos





```
for (x = 0; x < 1
repeat {"less
if (language
document.w
```



# Mais informações

- **Gamelan:** <http://www.gamelan.com>
- **JavaSoft:** <http://java.sun.com>
- **Java Beans:** <http://splash.javasoft.com/beans/>
- **JDBC:** <http://splash.javasoft.com/jdbc/>
- **Web Developer Java Resources:**  
<http://www.webdeveloper.com/frames/javalist.htm>
- **JavaWorld Magazine:** <http://www.javaworld.com>
- **Microsoft Java Corner:** <http://www.microsoft.com/java/>

