



## Applets

- Métodos da classe Applet
- Criando applets
- Passando parâmetros de uma página HTML
- Migrando de uma aplicação para uma applet
- Applets vs. Segurança

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Objetivos

- Migrar de uma aplicação para uma **applet**
- Escrever uma **applet** e integrar em uma página HTML
- Descrever que métodos são chamados pelo navegador e quando
- Passar parâmetros a uma **applet** à partir de uma página HTML
- Descrever as diferenças entre **applets** confiáveis e não-confiáveis

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Applet ?

- Pequena aplicação executada em uma janela de uma aplicação (browser/appletviewer)
- Tem por finalidade estender as funcionalidades de browsers, adicionando som, animação, etc., provenientes de fontes (URLs) locais ou remotas
- Cada página (arquivo .html) pode conter uma ou mais **applets**

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Applet?

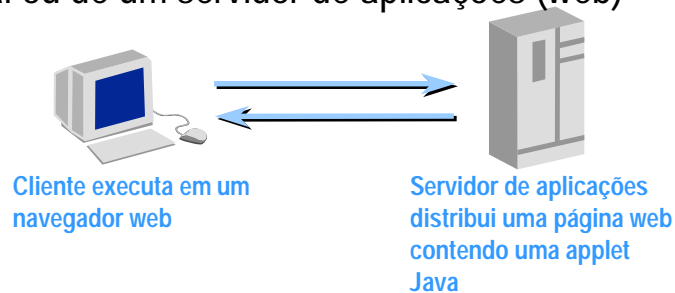
- A cada mudança de página as **applets** associadas são automaticamente destruídas
- Um navegador web que permite a execução de **applets** é dito **Java-enabled**.
- Uma página HTML contendo **applets** é dita **Java-powered**.

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Introdução

- Uma **applet** é incorporada em uma página web
- **Applets** podem ser carregadas a partir de um disco local ou de um servidor de aplicações (web)



Programação OO usando Java - Prof. Luciana Porcher Nedel



## Introdução

- O desenvolvimento ou não de programas em Java sob a forma de **applets** depende do objetivo da aplicação. Se por um lado a execução necessita de um navegador, por outro pode ser disponibilizada para execução via internet. Cabe ao projetista da aplicação definir a forma mais adequada para seu sistema.
- **Applets** sempre executam nos clientes web, nunca nos servidores. Por esta razão a carga das classes pode levar algum tempo. Para definir programas Java que rodem no servidor, veja como construir **servlets**.

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Características

### Aplicações

- programa autônomo
- executa em uma JVM autônoma
- ponto de entrada é o método main(), chamado pelo carregador de classes
- geralmente contém construtores
- acesso a todos os recursos do ambiente
- pode ser gráfica
- funcionalidades plenas: menus, diálogos, arquivos...

### Applets

- executada por um navegador
- o navegador fornece a JVM
- vários pontos de entrada são chamados pelo navegador como init()
- construtores tendem a ser vazios
- são sempre gráficas
- devem fornecer uma classe que estenda Applet
- funcionalidades limitadas por razões de segurança
- carregador de classes remotas

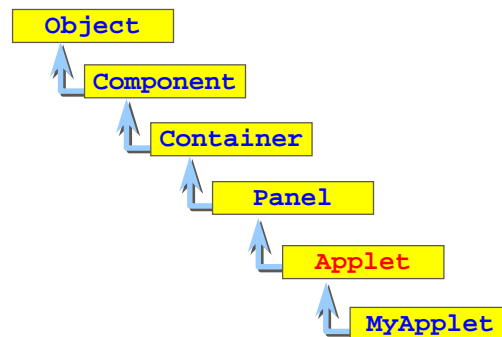
Programação OO usando Java - Prof. Luciana Porcher Nedel



## A classe Applet

- Para criar uma **applet**, cria-se uma sub-classe de Applet e redefine-se certos métodos:

- init()
- start()
- stop()
- destroy()
- paint()
- update()



Programação OO usando Java - Prof. Luciana Porcher Nedel



## A classe Applet

- Especialização de Panel: um container simples para exibir elementos gráficos
- Oferece um conjunto de métodos para:
  - iniciar/controlar/ terminar a execução de uma aplicação em um browser
  - buscar e executar componentes de som e imagem
- Segurança: fortes restrições a arquivos

Programação OO usando Java - Prof. Luciana Porcher Nedel



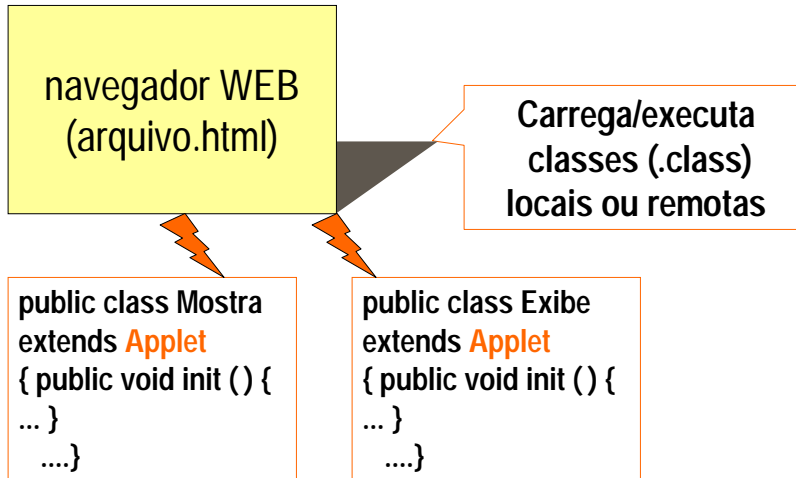
## Package java.applet

- Fornece as classes para:
  - criar **applets**
  - permitir que **applets** se comuniquem com seu contexto
- Contexto de uma **applet**:
  - uma aplicação que é responsável por carregar e executar **applets**.
    - Exemplo: navegadores e "appletviewer"
  - o código html que a contém e as demais **applets** contidas no mesmo documento

Programação OO usando Java - Prof. Luciana Porcher Nedel



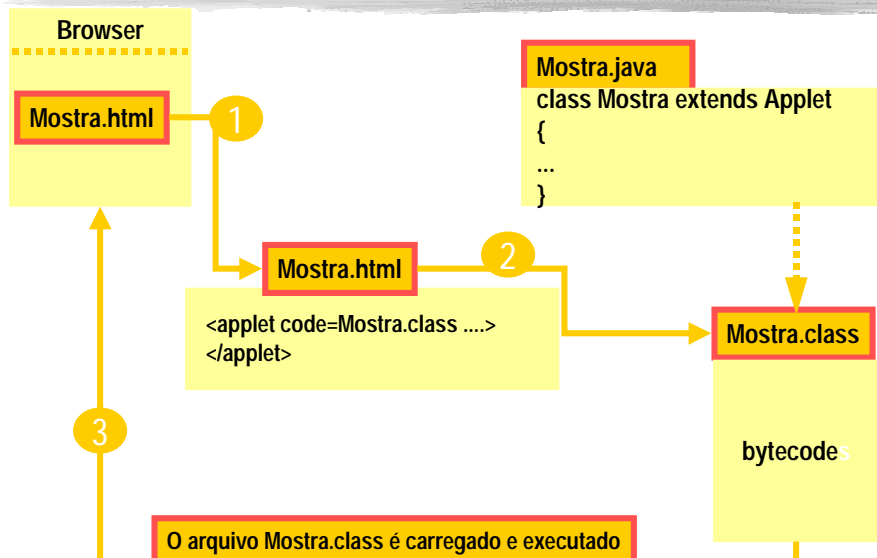
## Execução de applets



Programação OO usando Java - Prof. Luciana Porcher Nedel



## Execução em HTML



Programação OO usando Java - Prof. Luciana Porcher Nedel



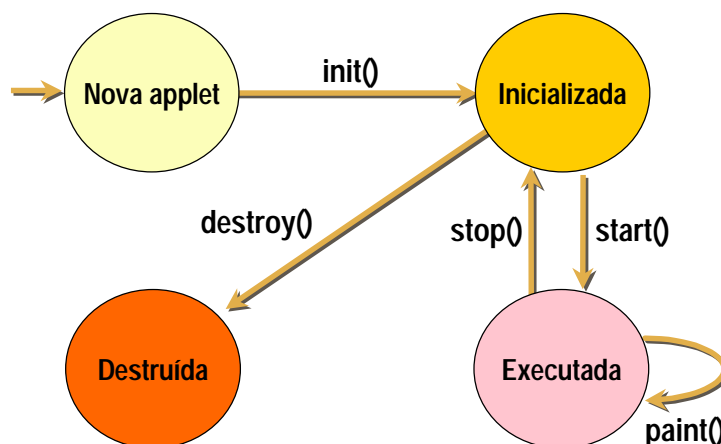
## Os 4 métodos do ciclo de vida da applet

- `init()` - inicialização; chamado quando a **applet** é inicialmente carregada
- `start()` - execução (animação); chamado após o `init()`
- `stop()` - interrupção; faz a **applet** parar a execução da animação, áudio ou vídeo
- `destroy()` - liberação de recursos; chamado quando o browser é fechado

Programação OO usando Java - Prof. Luciana Porcher Nedel



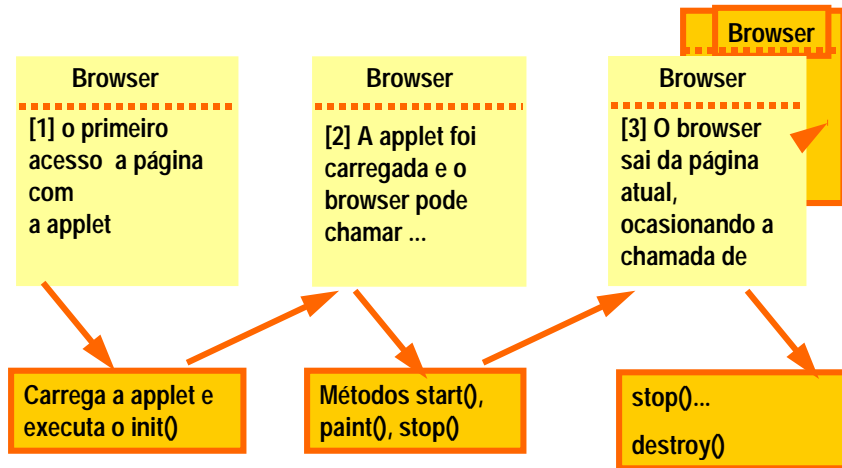
## Ciclo de vida de uma applet



Programação OO usando Java - Prof. Luciana Porcher Nedel



## Ciclos do processo de execução



Programação OO usando Java - Prof. Luciana Porcher Nedel



## Esqueleto de uma applet

```
public class MinhaApplet extends Applet {
    public void init(){
        // Inicializa a applet
    }
    public void start(){
        // Dispara a execução de uma applet
    }
    public void stop(){
        // Pára a execução de uma applet
    }
    public void destroy(){
        // Destrói definitivamente uma applet
    }
    public void paint(Graphics g){
        ...
    }
}
```

Programação OO usando Java - Prof. Luciana Porcher Nedel





## Implementando uma applet

- Toda **applet** é uma aplicação gráfica. Não existem **applets** “modo texto”.
- A principal diferença entre uma “Java application” e uma “applet” é o fato de que a janela base da aplicação será derivada a partir da classe Applet e não a partir da classe Frame.
- Além disso, a parte da aplicação que instancia a classe Applet e relaciona-a com o browser é padrão e, portanto, não precisa ser descrita. Desta forma, **applets** não terão a função “main()”.
- Veja o exemplo:

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Implementando uma applet

```
import java.applet.*;
import java.awt.*;

public class Exemplo extends Applet{
    private int x,y,tam;

    public void init(){           << Faz o papel do construtor
        x = 10; y = 10; tam = 0;
    }
    public void start(){         << Ativado logo após init e toda a vez que a applet
        tam += 5;                for revisitado
    }
    public void paint(Graphics g){
        g.setColor(new Color(200,123,212));
        g.fillRect(x,y,tam,10);
    }
    public void stop(){         << Ativado sempre que a applet pode parar de
    }                             executar (não está visível)
}
```

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Inserindo uma applet em uma página

- Para que possa ser executada, uma **applet** deve ser associada a uma página www através de código HTML. Veja o exemplo a seguir:

```
<HTML>
<HEAD>
<TITLE>Applet title</TITLE>
</HEAD>
<BODY>
<H1>First Heading</H1>
<HR>
<APPLET CODE="Exemplo.class" WIDTH=120 HEIGHT=150></APPLET>
<HR>
</BODY>
</HTML>
```

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Inserindo uma applet em uma página

- **Atenção:** normalmente os navegadores estão configurados para carregar cada **applet** apenas na primeira vez que uma página é acessada.
- Logo, se a **applet** for recompilada nesse meio tempo, será necessário forçar a sua recarga (não adianta fazer apenas o "reload" da página).

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Exemplo de applet

```
import java.applet.Applet;
import java.awt.*;
// Exibe um ditado de Confucio do livro:
// Java Software Solutions:J. Lewis e W.
// Loftus
public class Confucio extends Applet{
    public void paint(Graphics pagina){
        pagina.drawString("Esqueça desaforos,
nunca esqueça gentilezas.", 30,30);
        pagina.drawString("----Confúcio",
70,70);
    } //ditado
} // classe Confucio
```

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Arquivo HTML associado: nome.html

```
<HTML>
<BODY>
  <APPLET CODE =
    nomeclasse.class
    WIDTH =...
    HEIGHT= ...>
</APPLET>
</BODY>
</HTML>
```

```
<! Confucio.html>
<HTML>
<HEAD>
<TITLE> CONFUCIO</TITLE>
</HEAD>
<BODY>
<H3>
<APPLET CODE="Confucio.class"
WIDTH=300 HEIGHT=150>
</APPLET>
<HR>
</BODY>
</HTML>
```

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Recuperando arquivos de imagens

- Carregando um objeto imagem do diretório de onde foi **lido o .html**:

```
Image img, img2;  
img=getImage(getDocumentBase(), "lenna.jpg");
```

- Carrega um objeto imagem do diretório de onde foi **carregado o .class**:

```
img2=getImage(getCodeBase(), "mandril.jpg");  
| Exemplo (ImageDisplayer.java)  
| Exemplo (ImageDisplayer2.java)
```

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Criando applets: exibindo uma imagem

```
import java.applet.*;  
import java.awt.*;  
public class Imagem1 extends Applet  
{  
    Image imagem;  
    public void init()  
    {  
        imagem=getImage(getDocumentBase(), "mandril.jpg");  
    }  
    public void paint(Graphics g)  
    {  
        g.drawImage(imagem,0,0,this);  
        g.drawString("Macaco Mandril", 50, 25);  
    }  
}
```

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Parametrização em uma página web

```
< APPLET
  CODE = appletFile // Obrigatório. Fornece o nome da classe que contém a applet
  WIDTH = pixels // Obrigatório. Fixa o tamanho inicial da applet em pixels
  HEIGHT = pixels
  [ CODEBASE = codebaseURL ] // Opcional. Define a base URL da classe do applet
  [ ALT = alternateText ] // Permite exibir uma mensagem para o browser
  [ NAME = appletInstanceName ] // Opcional. Fornece a instância de uma applet em
                                // uma página HTML, de modo a estabelecer uma
                                // comunicação entre applets
  [ ALIGN = alignment ] // Posicionamento de uma applet em uma página HTML. Os
                        // valores permitidos são: left, right, top, texttop, middle,
                        // absmiddle, baseline, bottom, absbottom.
  [ VSPACE = pixels ] // Número de pixels entre a applet e o resto da página HTML
  [ HSPACE = pixels ]
>
[ < PARAM NAME = appletParameter1 VALUE = value > ]
// Único modo de passar argumentos a uma applet
[ < PARAM NAME = appletParameter2 VALUE = value > ]
```

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Método getParameter()

- `getParameter()`: retorna o valor do parâmetro indicado existente no documento base (.html)

- `public String getParameter(String name)`

- Exemplo

arquivo .html:

```
<APPLET CODE="Confucio.class" ...>
<PARAM NAME="title" VALUE="Dito de Confucio">
</APPLET>
```

```
mtitulo = getParameter("title");
```

arquivo .java

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Um html -> várias applets

```
<html>
<body>
<!CENTER>
<applet CODE="Confucio.class" WIDTH="415"
HEIGHT="83">
<param NAME="title" VALUE="Panel X Applet:1">
<applet CODE="MinhaApplet.class" WIDTH="483"
HEIGHT="104">
<param NAME="title" VALUE="Panel X Applet:2"></applet>
</body>
</html>
```

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Outros métodos da classe Applet

- `resize()`: redefine o tamanho da **applet**;
  - `public void resize(int width,int height)`
- `getImage()`: retorna um objeto `Image` que pode ser exibido na tela pelo método `paint()`
  - `public Image getImage(URL url)`
- `newAudioClip()`: busca um clip de áudio que pode ser executado pelo método `play()`
  - `public static final AudioClip newAudioClip(URL url)`

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Métodos URL

`getCodeBase()`: retorna a URL base da classe

```
public URL getCodeBase()
```

`getDocumentBase()`: retorna a URL base do HTML

```
public URL getDocumentBase()
```

Exemplo:

```
Image img, img2;
```

```
img=getImage(getDocumentBase(), "tulipas.gif");
```

```
img2=getImage(getCodeBase(), "coracao.gif");
```

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Transformando aplicações em applets

### ■ Passo 1: Eliminar o `main()`

- usando applets o browser cria o objeto da classe especificada no TAG :

- `<APPLET CODE = ... >`

- O tamanho da applet é definido por `WIDTH` e `HEIGHT`

### ■ Passo 2: Derivar a classe de `Applet` (public):

- `public class MinhaClasse extends Applet`

### ■ Passo 3: Substituir o construtor pelo método `init()`

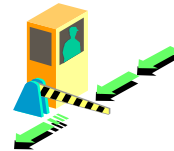
Programação OO usando Java - Prof. Luciana Porcher Nedel



## Applets vs. Segurança

- O que uma **applet** Java pode ou não, é determinado pelo gerente de segurança do browser e pela segurança do sistema.
- Limitações default: uma **applet** não pode
  - acessar arquivos na máquina local
  - chamar outro programa na máquina local
  - se comunicar com nenhuma outra máquina a não ser a que contém a página HTML em questão
- Estas limitações podem ser ocasionalmente autorizadas através de um arquivo especial (**policy file**)

Define o que é permitido para um código-fonte particular

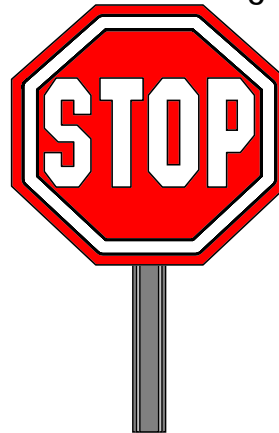


Programação OO usando Java - Prof. Luciana Porcher Nedel



## Parada para exercícios

- Abrir os seguintes arquivos em um navegador:
  - Confucio.html
  - MinhaApplet.html



Programação OO usando Java - Prof. Luciana Porcher Nedel





## Exercício

Carregar e executar, a partir do diretório indicado, os seguintes arquivos: MinhaApplet.\* e Confucio.\*. A seguir, fazer alguma alteração (cor, tamanho, texto, etc) nos fontes das **applets** e gerar um arquivo .html para exibir pelo menos três **applets**.

Usar duas formas de visualização: com appletviewer e com um browser(Netscape/Explorer)

Exemplo de execução com Appletviewer:

```
:jdk1.2.1\bin\javac MinhaApplet.java
```

```
:jdk1.2.1\bin\appletviewer MinhaApplet.html
```

Programação OO usando Java - Prof. Luciana Porcher Nedel

(continua...)



## Exercício

■ Importante: o ambiente de execução de Java fica bloqueado aguardando o final da execução. A janela AppletViewer deve ser fechada (X) para terminar a execução.

■ A execução com Browser pode ser feita:

- Clicando o nome/ícone do arquivo .html
- Usando a opção Abrir Página (local) do browser.



Programação OO usando Java - Prof. Luciana Porcher Nedel



## Java e a biblioteca de classes Swing

```
import javax.swing.JApplet; // importa a classe JApplet
import java.awt.Graphics; // importa a classe Graphics

public class WelcomeApplet extends JApplet {
    public void paint( Graphics g )
    {
        g.drawString( "Welcome to Java Programming!", 25, 25 );
    }
}
```

- Outra maneira de programar **applets** é usando os componentes da **Swing** (uma extensão à AWT), que serão abordados em profundidade mais tarde...

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Desenvolvimento de applets

- Tratamento de eventos
  - procedimentos em resposta a ações do usuário
- Classes do package de classes de interface
  - elementos gráficos básicos como botões, scrollbars, painéis, frames, canvases, etc.
- Classes do AWT
  - elementos gráficos básicos como botões, scrollbars, painéis, frames, canvases, etc
- Classes de java Swing
  - reúnem elementos básicos de forma mais encapsulada
  - Swing praticamente substitui AWT, pois fornece componentes de interface mais poderosos

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Desenvolvimento de applets

- Se houver geração de desenhos
  - Graphics (AWT)
  - Java3D

→ *observação: estes requisitos são também os de desenvolvimento de aplicações em Java*

Programação OO usando Java - Prof. Luciana Porcher Nedel



## Applets - Programação Orientada a Eventos

- Applets, sendo programas gráficos e interativos, são direcionadas a eventos.
  - Eventos correspondem à ocorrência de algum fato, o qual deve (ou não) desencadear um procedimento específico
- Não há, em princípio, uma seqüência para a execução de procedimentos.
  - Procedimentos são acionados dependendo de ações do usuário.

Programação OO usando Java - Prof. Luciana Porcher Nedel