

Módulo 2

Introdução à Programação II



Lição 11

Applets

Versão 1.0 - Mar/2007

Autor

Rebecca Ong

Equipe

Joyce Avestro
 Florence Balagtas
 Rommel Feria
 Rebecca Ong
 John Paul Petines
 Sun Microsystems
 Sun Philippines

Necessidades para os Exercícios**Sistemas Operacionais Suportados****NetBeans IDE 5.5** para os seguintes sistemas operacionais:

- Microsoft Windows XP Professional SP2 ou superior
- Mac OS X 10.4.5 ou superior
- Red Hat Fedora Core 3
- Solaris™ 10 Operating System (SPARC® e x86/x64 Platform Edition)

NetBeans Enterprise Pack, poderá ser executado nas seguintes plataformas:

- Microsoft Windows 2000 Professional SP4
- Solaris™ 8 OS (SPARC e x86/x64 Platform Edition) e Solaris 9 OS (SPARC e x86/x64 Platform Edition)
- Várias outras distribuições Linux

Configuração Mínima de Hardware**Nota:** IDE NetBeans com resolução de tela em 1024x768 pixel

Sistema Operacional	Processador	Memória	HD Livre
Microsoft Windows	500 MHz Intel Pentium III workstation ou equivalente	512 MB	850 MB
Linux	500 MHz Intel Pentium III workstation ou equivalente	512 MB	450 MB
Solaris OS (SPARC)	UltraSPARC II 450 MHz	512 MB	450 MB
Solaris OS (x86/x64 Platform Edition)	AMD Opteron 100 Série 1.8 GHz	512 MB	450 MB
Mac OS X	PowerPC G4	512 MB	450 MB

Configuração Recomendada de Hardware

Sistema Operacional	Processador	Memória	HD Livre
Microsoft Windows	1.4 GHz Intel Pentium III workstation ou equivalente	1 GB	1 GB
Linux	1.4 GHz Intel Pentium III workstation ou equivalente	1 GB	850 MB
Solaris OS (SPARC)	UltraSPARC IIIi 1 GHz	1 GB	850 MB
Solaris OS (x86/x64 Platform Edition)	AMD Opteron 100 Series 1.8 GHz	1 GB	850 MB
Mac OS X	PowerPC G5	1 GB	850 MB

Requerimentos de Software

NetBeans Enterprise Pack 5.5 executando sobre Java 2 Platform Standard Edition Development Kit 5.0 ou superior (JDK 5.0, versão 1.5.0_01 ou superior), contemplando a Java Runtime Environment, ferramentas de desenvolvimento para compilar, depurar, e executar aplicações escritas em linguagem Java. Sun Java System Application Server Platform Edition 9.

- Para **Solaris, Windows, e Linux**, os arquivos da JDK podem ser obtidos para sua plataforma em <http://java.sun.com/j2se/1.5.0/download.html>
- Para **Mac OS X**, Java 2 Platform Standard Edition (J2SE) 5.0 Release 4, pode ser obtida diretamente da Apple's Developer Connection, no endereço: <http://developer.apple.com/java> (é necessário registrar o download da JDK).

Para mais informações: <http://www.netbeans.org/community/releases/55/relnotes.html>

Colaboradores que auxiliaram no processo de tradução e revisão

Alexandre Mori	Hugo Leonardo Malheiros Ferreira	Mauro Regis de Sousa Lima
Alexis da Rocha Silva	Ivan Nascimento Fonseca	Namor de Sá e Silva
Aline Sabbatini da Silva Alves	Jacqueline Susann Barbosa	Néres Chaves Rebouças
Allan Wojcik da Silva	Jader de Carvalho Belarmino	Nolyanne Peixoto Brasil Vieira
André Luiz Moreira	João Aurélio Telles da Rocha	Paulo Afonso Corrêa
Andro Márcio Correa Louredo	João Paulo Cirino Silva de Novais	Paulo José Lemos Costa
Antonie de Assis Lima	João Vianney Barrozo Costa	Paulo Oliveira Sampaio Reis
Antonio Jose R. Alves Ramos	José Augusto Martins Nieviadonski	Pedro Antonio Pereira Miranda
Aurélio Soares Neto	José Leonardo Borges de Melo	Pedro Henrique Pereira de Andrade
Bruno da Silva Bonfim	José Ricardo Carneiro	Renato Alves Félix
Bruno dos Santos Miranda	Kleberth Bezerra G. dos Santos	Renato Barbosa da Silva
Bruno Ferreira Rodrigues	Lafaiete de Sá Guimarães	Reydersen Magela dos Reis
Carlos Alberto Vitorino de Almeida	Leandro Silva de Moraes	Ricardo Ferreira Rodrigues
Carlos Alexandre de Sene	Leonardo Leopoldo do Nascimento	Ricardo Ulrich Bomfim
Carlos André Noronha de Sousa	Leonardo Pereira dos Santos	Robson de Oliveira Cunha
Carlos Eduardo Veras Neves	Leonardo Rangel de Melo Filardi	Rodrigo Pereira Machado
Cleber Ferreira de Sousa	Lucas Mauricio Castro e Martins	Rodrigo Rosa Miranda Corrêa
Cleyton Artur Soares Urani	Luciana Rocha de Oliveira	Rodrigo Vaez
Cristiano Borges Ferreira	Luís Carlos André	Ronie Dotzlaw
Cristiano de Siqueira Pires	Luís Octávio Jorge V. Lima	Rosely Moreira de Jesus
Derlon Vandri Aliendres	Luiz Fernandes de Oliveira Junior	Seire Pareja
Fabiano Eduardo de Oliveira	Luiz Victor de Andrade Lima	Sergio Pomeranclum
Fábio Bombonato	Manoel Cotts de Queiroz	Silvio Sznifer
Fernando Antonio Mota Trinta	Marcello Sandi Pinheiro	Suzana da Costa Oliveira
Flávio Alves Gomes	Marcelo Ortolan Pazzetto	Tásio Vasconcelos da Silveira
Francisco das Chagas	Marco Aurélio Martins Bessa	Thiago Magela Rodrigues Dias
Francisco Marcio da Silva	Marcos Vinicius de Toledo	Tiago Gimenez Ribeiro
Gilson Moreno Costa	Maria Carolina Ferreira da Silva	Vanderlei Carvalho Rodrigues Pinto
Givailson de Souza Neves	Massimiliano Girolidi	Vanessa dos Santos Almeida
Gustavo Henrique Castellano	Mauricio Azevedo Gamarra	Vasti Mendes da Silva Rocha
Hebert Julio Gonçalves de Paula	Mauricio da Silva Marinho	Wagner Eliezer Roncoletta
Heraldo Conceição Domingues	Mauro Cardoso Mortoni	

Auxiliadores especiais

Revisão Geral do texto para os seguintes Países:

- **Brasil** – Tiago Flach
- **Guiné Bissau** – Alfredo Cá, Bunene Sisse e Buon Olossato Quebi – ONG Asas de Socorro

Coordenação do DFJUG

- **Daniel deOliveira** – JUGLeader responsável pelos acordos de parcerias
- **Luci Campos** - Idealizadora do DFJUG responsável pelo apoio social
- **Fernando Anselmo** - Coordenador responsável pelo processo de tradução e revisão, disponibilização dos materiais e inserção de novos módulos
- **Regina Mariani** - Coordenadora responsável pela parte jurídica
- **Rodrigo Nunes** - Coordenador responsável pela parte multimídia
- **Sérgio Gomes Veloso** - Coordenador responsável pelo ambiente JEDI™ (Moodle)

Agradecimento Especial

John Paul Petines – Criador da Iniciativa JEDI™

Rommel Faria – Criador da Iniciativa JEDI™

1. Objetivos

Applets é uma das variedades mais interessantes de Java. Refere-se a aplicativos que são executados por meio de um navegador de Internet. Iremos conhecer mais sobre como criar applets nessa lição.

Ao final desta lição, o estudante será capaz de:

- Definir o que é uma *applet*
- Criar sua própria *applet*
- Explicar o ciclo de vida de uma *applet*
- Utilizar os métodos das *applets*
 - Métodos para mostrar a *applet*
 - Métodos para reproduzir clipes de áudio
- Utilizar a tag applet de HTML

2. Criando *Applets*

Uma *applet* é um tipo especial de classe Java que é executado pela Internet. É normalmente executada por um navegador de Internet, tipo: Netscape Navigator, Mozilla ou Microsoft Internet Explorer. No entanto, comparada com outras aplicações Java, applets não tem acesso ao computador na qual está sendo executada por razões de segurança. Isso faz com que *applets* sejam bastante restritas se comparadas com outras aplicações Java.

2.1. *Hello World Applet*

A classe *Applet* é uma subclasse da classe *Panel* definida na AWT. A melhor forma de entender como criar uma *applet*, é através de exemplos. Então, aqui está uma simples *applet* que mostra a mensagem "Hello world!".

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

public class AppletDemo extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello world!", 80, 25);
    }
}
```

Depois da compilação, tente rodar esse exemplo usando o comando: **java AppletDemo**. O que acontece? Lembre-se que as *applets* são aplicações Java especiais. Elas não são executadas utilizando o comando **java**. Em vez disso, a *applet* é executada por um navegador de internet ou através de um aplicativo denominado *appletviewer*. Para executar a *applet* através de um navegador de Internet, crie uma página HTML utilizando a tag *Applet* da linguagem HTML. Para isso, crie um arquivo com a extensão .html contendo o seguinte código:

```
<applet code="AppletDemo" width=300 height=100>
</applet>
```

A tag no exemplo dado, indica que uma *applet* seja criada com a largura de 300 pixels e altura de 100 pixels. Então, o método `drawString` desenha a frase "Hello world!" na posição em pixels (80,25) contando para baixo e depois à direita.

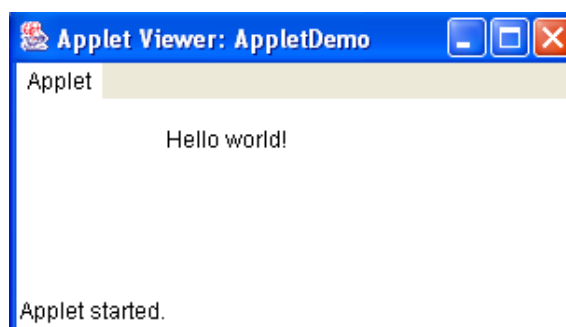


Figura 1: exemplo de Applet

Outra forma de executar uma *applet* é através do aplicativo *appletviewer*. Simplesmente siga essa sintaxe:

appletviewer <nome do arquivo java>

Por exemplo, para executar o exemplo da *applet*, digite em uma janela de console:

```
appletviewer AppletDemo.java
```

Ao se criar uma *applet*, é necessário estender a classe *Applet*. Como foi mencionado previamente, essa classe é encontrada no pacote *java.applet*. Portanto, é extremamente necessário importar o pacote *java.applet*. Além disso, foi mencionado anteriormente que a classe *Applet* é uma subclasse da classe *Panel*. Isso implica que alguns métodos da classe *Applet* são encontradas na classe *Panel*. Para acessar os métodos ou campos na classe *Panel* ou outras classes ancestrais, será necessário importar o pacote *java.awt*.

3. Métodos da *Applet*

Esta secção demonstrará alguns métodos de *Applets* extremamente interessantes

3.1. O ciclo de vida de uma *Applet*

Em vez de começar a execução no método *main* como uma aplicação Java típica, o navegador ou o aplicativo appletviewer com a *applet* através dos seguintes métodos:

1. *init()*

init é o primeiro método chamado. Ele é chamado assim que a *applet* for carregada.

2. *start()*

Após chamar o método *init*, *start* é o próximo método chamado. Ele é chamado cada vez que a página HTML da *applet* é exibida. Quando a *applet* é reexibido, a execução começa com esse método.

3. *stop()*

Quando o navegador de Internet sai da página HTML do *applet*, esse método é chamado para informar ao *applet* que ele deve interromper sua execução.

4. *destroy()*

Esse método é chamado quando o *applet* precisa ser completamente removido da memória. O método *stop* é sempre chamado antes que esse método seja chamado.

Quando criamos *applets*, pelo menos alguns desses métodos devem ser *overriden*. O exemplo seguinte de *applet* realiza este processo:

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

public class LifeCycleDemo extends Applet {
    String msg = "";
    public void init() {
        msg += "initializing... ";
        repaint();
    }
    public void start() {
        msg += "starting... ";
        repaint();
    }
    public void stop() {
        msg += "stopping... ";
        repaint();
    }
    public void destroy() {
        msg += "preparing for unloading... ";
        repaint();
    }
    public void paint(Graphics g) {
        g.drawString(msg, 15, 15);
    }
}
```

A seguir temos um exemplo de um documento HTML embutido com um *applet* LifeCycleDemo.

```
<HTML>
<TITLE>Life Cycle Demo</TITLE>
```

```
<applet code="LifeCycleDemo" width=300 height=100>
</applet>
</HTML>
```

3.2. Método *paint*

Outro importante método é o *paint*, que a classe *Applet* herda de sua classe ancestral *Component*. Este método é chamado toda vez que a saída da *applet* precisa ser redesenhada. Um exemplo deste comportamento é quando a *applet*, escondido por outra janela e fica visível novamente. Esse método pode ser substituído quando for necessário customizar a aparência da *applet*. No exemplo para a classe *AppletDemo*, onde é mostrada a mensagem "Hello world!" no fundo depois de ter substituído o método *paint*.

3.3. Método *showStatus*

Applet tem uma janela de status, que informa o que ela está fazendo no momento. Se for necessário mostrar alguma mensagem na janela de status, chame o método *showStatus*. O exemplo a seguir é idêntico ao *AppletDemo*, entretanto com comandos adicionais que modificam o conteúdo da janela de status.

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

public class AppletDemo extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello world!", 80, 25);
        showStatus("This is an important information.");
    }
}
```

A seguir, o resultado do exemplo:

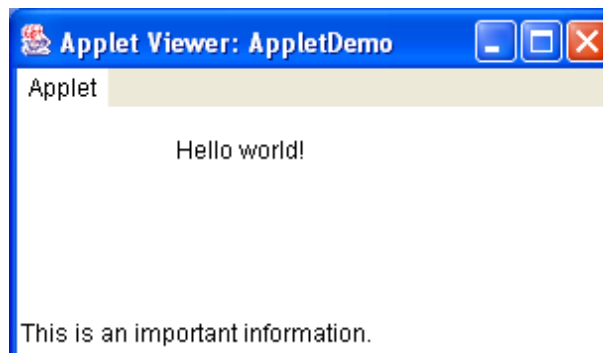


Figura 2: exemplo *showStatus()*

3.4. Reproduzindo clipes de Áudio

Applets também são providas de métodos que permitem reproduzir arquivos de áudio. Reproduzir clipes de áudio em uma *applet* envolve as seguintes etapas:

1. Obter o clipe de áudio usando o método *getAudioClip*.
2. Para reproduzir o clipe de áudio, use o método *play* ou o método *loop* no objeto do clipe de áudio.
3. O método *play* permite a reprodução única do áudio, enquanto que o método *loop* reproduz o áudio continuamente, parando apenas quando o método *stop* for chamado.

O próximo exemplo reproduz continuamente um áudio até que o método *stop* da *applet* seja

acionado.

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

public class AudioApplet extends Applet {
    AudioClip ac;
    public void init() {
        try {
            /* clipe de audio foi salvo no mesmo diretório do código java */
            /* spaceMusic foi baixado de java.sun.com */
            ac = getAudioClip(getCodeBase(), "spaceMusic.au");
            ac.loop();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
    public void stop() {
        ac.stop();
    }
    public void paint(Graphics g) {
        g.drawString("Playing space music!", 80, 25);
    }
}
```

4. *Applet* HTML Tags

Nos exemplos anteriores, foi mostrado como a tag HTML de *applet* é utilizada nas páginas HTML ou no fonte Java. Agora, veremos a forma mais completa da tag HTML para *applet*.

```
<APPLET
  [CODEBASE = codebaseURL]
  CODE = appletFile
  [ATL = alternateText]
  [NAME = appletInstanceName]
  WIDTH = widthInPixels HEIGHT = heightInPixels
  [ALIGN = alignment]
  [VSPACE = vspaceInPixels] [HSPACE = hspaceInPixels]
>
[<PARAM NAME = parameterName1 VALUE = parameterValue1>]
[<PARAM NAME = parameterName2 VALUE = parameterValue2>]
...
[<PARAM NAME = parameterNameN VALUE = parameterValueN>]
[HTML que será mostrado na ausência do Java]
</APPLET>
```

Palavras-chave de tags HTML para Applets	
CODEBASE	
Diretório onde a classe <i>applet</i> está localizada. Por padrão, utiliza o mesmo diretório URL do documento HTML.	
CODE	
Nome do arquivo que contém o código <i>applet</i> . Com ou sem as extensões .java ou .class.	
ALT	
Texto mostrado quando o navegador conhece tags <i>applet</i> mas não consegue executá-la. Pode acontecer se o Java do navegador estiver desabilitado.	
NAME	
Nome da <i>applet</i> . Usado para permitir que outras <i>applets</i> se comuniquem com o applet usando esse nome.	
WIDTH, HEIGHT	
Largura e altura da janela da <i>applet</i> . Especificado em pixels.	
ALIGN	
Alinhamento ou posicionamento da applet. Pode receber diversos valores "left", "right", "top", "bottom", "middle", "baseline", "texttop", "absmiddle", ou "absbottom". O posicionamento padrão depende do ambiente.	
VSPACE, HSPACE	
Espaço acima e abaixo (VSPACE) e o dos lados (HSPACE) do applet.	
PARAM NAME, VALUE	
Serve para especificar parâmetros que podem ser passados para o applet; applets podem chamar o método <i>getParameter(String paramName)</i> .	

Tabela 1: tags HTML para Applets

Observação:

- **top** – topo do applet alinhado com o item mais alto da mesma linha.

- **bottom, baseline** – parte inferior do applet alinhado com a parte inferior de outros itens da mesma linha.
- **middle** – meio do applet alinhado com a parte inferior de outros componentes na mesma linha.
- **texttop** – topo do applet alinhado com o topo do texto mais alto da mesma linha .
- **absmiddle** – meio do applet alinhado com o meio vertical de outros componentes da mesma linha.
- **absbottom** – parte inferior do applet alinhado com a parte inferior de outros componentes da mesma linha.

O exemplo abaixo demonstra como acessar parâmetros especificados na tag HTML.

```
import java.awt.*;  
import java.applet.*;  
  
public class ParamDemo extends Applet {  
    public void paint(Graphics g) {  
        g.drawString(getParameter("myParam"), 80, 25);  
    }  
}
```

O resultado dessa classe é o mesmo da *AppletDemo*. Logo abaixo vemos a tela com o resultado esperado:

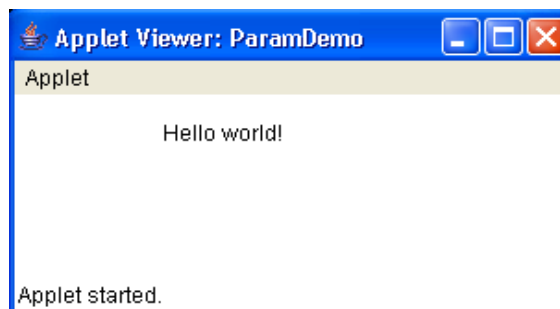


Figura 3: Exemplo de execução do ParamDemo

Parceiros que tornaram JEDI™ possível



Instituto CTS

Patrocinador do DFJUG.

Sun Microsystems

Fornecimento de servidor de dados para o armazenamento dos vídeo-aulas.

Java Research and Development Center da Universidade das Filipinas
Criador da Iniciativa JEDI™.

DFJUG

Detentor dos direitos do JEDI™ nos países de língua portuguesa.

Banco do Brasil

Disponibilização de seus *telecentros* para abrigar e difundir a Iniciativa JEDI™.

Politec

Suporte e apoio financeiro e logístico a todo o processo.

Borland

Apoio internacional para que possamos alcançar os outros países de língua portuguesa.

Instituto Gaudium/CNBB

Fornecimento da sua infra-estrutura de hardware de seus servidores para que os milhares de alunos possam acessar o material do curso simultaneamente.