

Programação Visual Forms com Java e Animações em C

Prof. Me. Ederson Cichaczewski

Conversa Inicial

- Tema 1
 - Conceitos da Linguagem Java
- Tema 2
 - Java IDE e Interface Visual

- Tema 3
 - Projetos comJava
- Tema 4
 - Biblioteca GráficaOpenGL com C
- Tema 5
 - Projetos comOpenGL

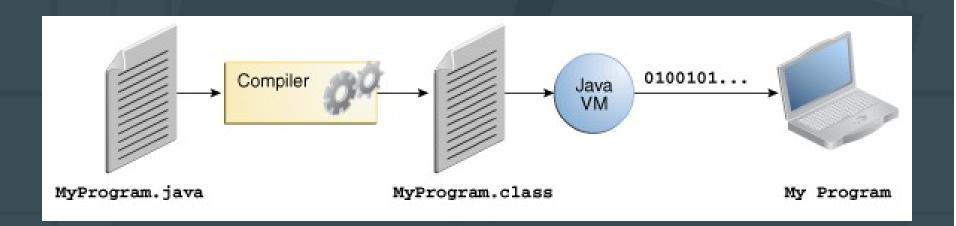
Contextualizando

- Como são desenvolvidos os aplicativos com forms em Java?
- Ferramenta: NetBeans IDE
- Recursos OpenGL em Linguagem C

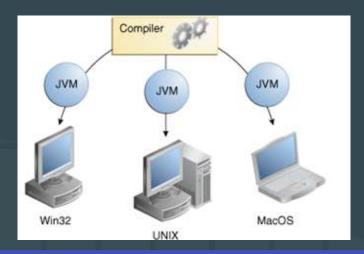
Tema 1 Conceitos da Linguagem Java

- Linguagem Java
 - Alto nível
 - Independe de arquitetura
 - Orientada ao objetos
 - Multitarefa etc.





Máquina Virtual: Java Virtual Machine (JVM)

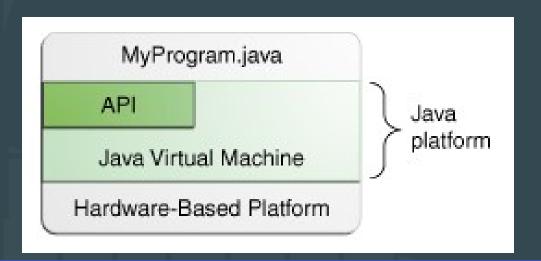


Pacotes

JRE (Java Runtime **Environment):** destinado a usuários, inclui a JVM e as bibliotecas (APIs) que permitem rodar as aplicações Java

Development Kit):
destinado a
desenvolvedores,
permite compilar
aplicações Java

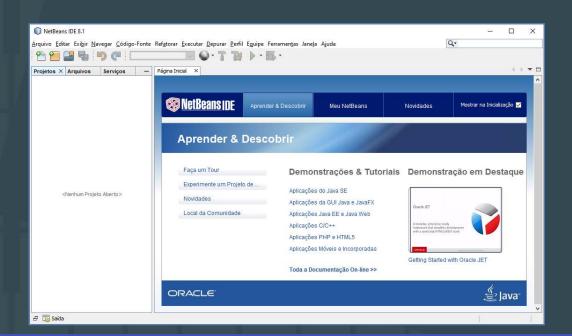
Plataforma Java



Tema 2 Java IDE e **Interface Visual**

Ferramenta: NetBeans IDE

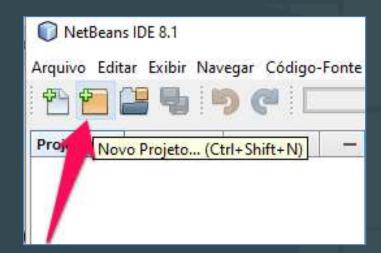
Gratuita e ideal para forms em Java, com API Swing integrada

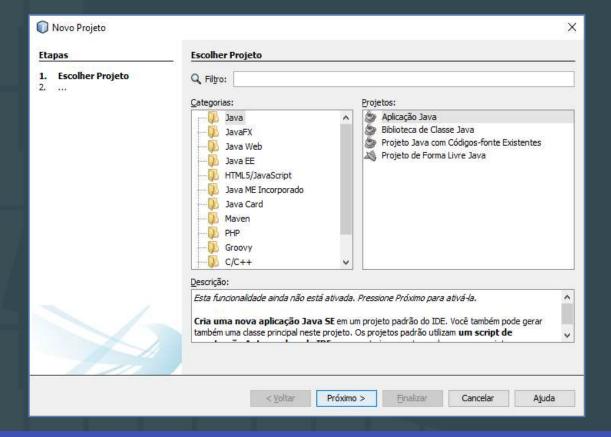


12 67

Novo projeto - Aplicação Java

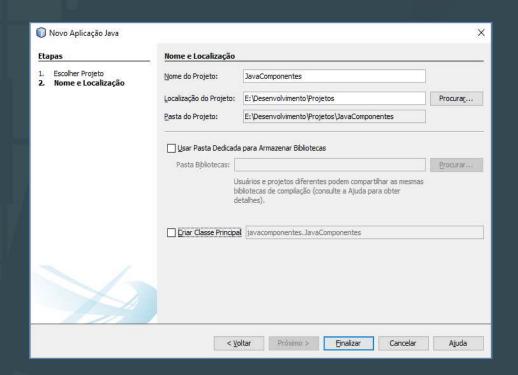
13 67 ×





Próximo passo: Nome do Projeto

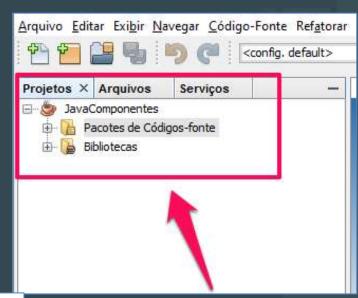
Desmarcar a opção Criar Classe Principal

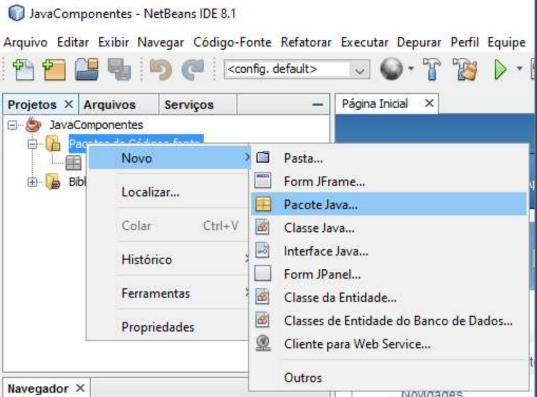


Finalizar....

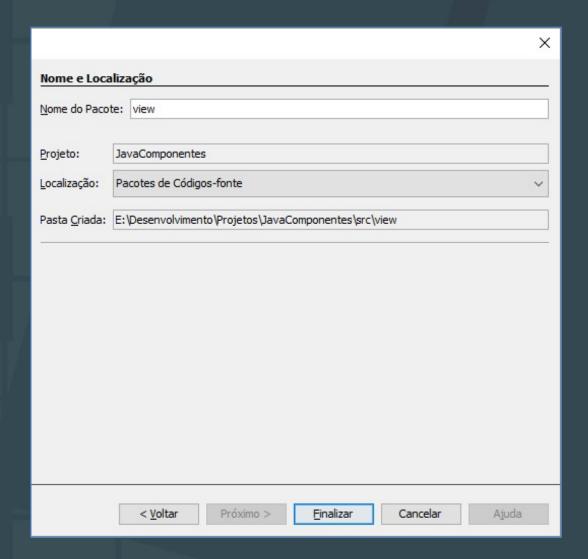
Aba Projetos

15 67 Criar um Pacote de Código Java



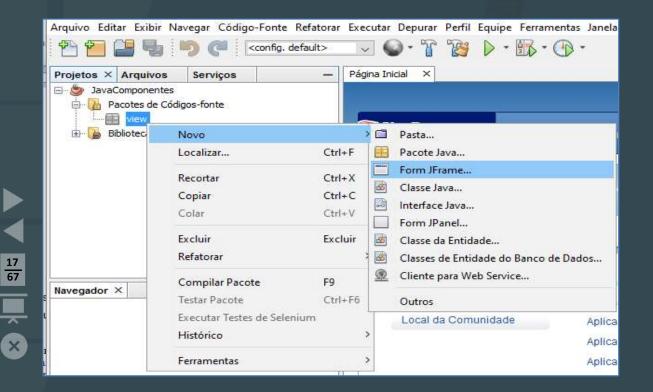


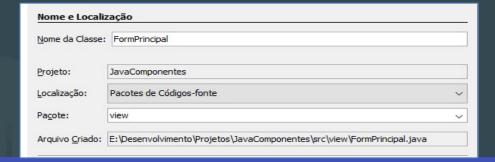
Novo Pacote: view



• Finalizar...

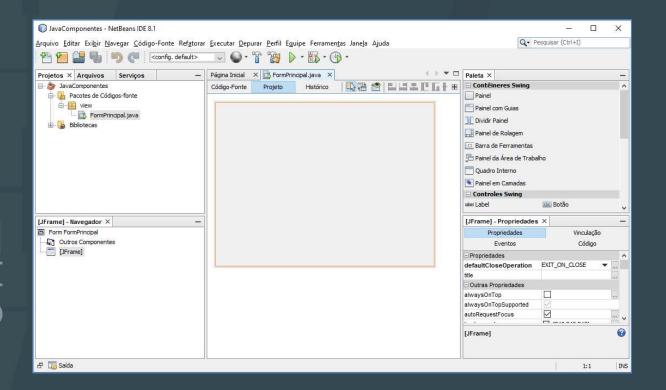
Novo Componente: Form JFrame





Visão geral do ambiente

18 67



Esquerda: estrutura do projeto Centro: interface visual Direita: componentes e propriedades

Componentes

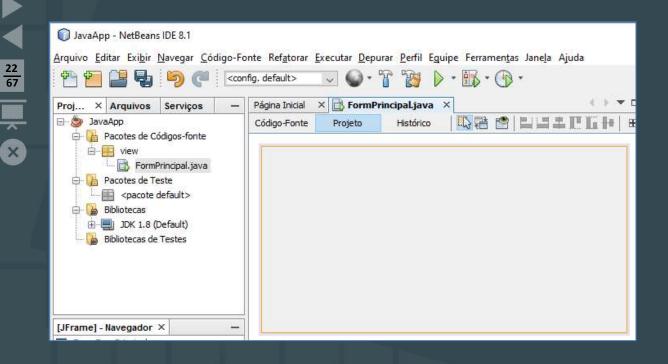
Exemplo com: label, campo de texto, campo de senha, área de texto, caixa de seleção, botão de seleção, lista, botão...



Tema 3 Projetos com Java

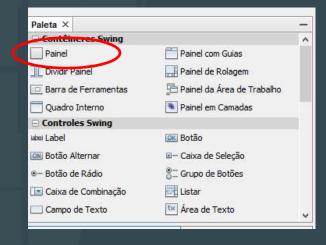
Aplicação Form JFrame Swing

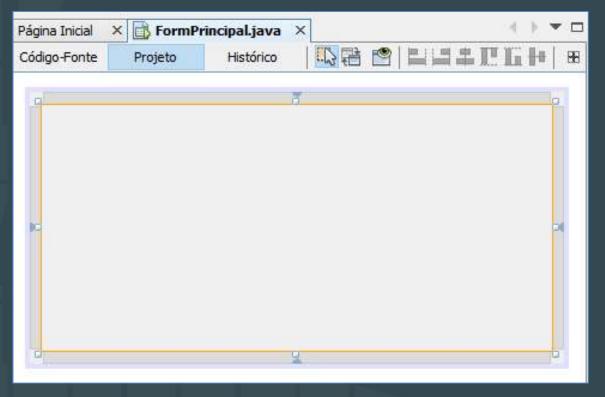
Realizar os passos de criação de projeto Java, novo pacote, novo form JFrame



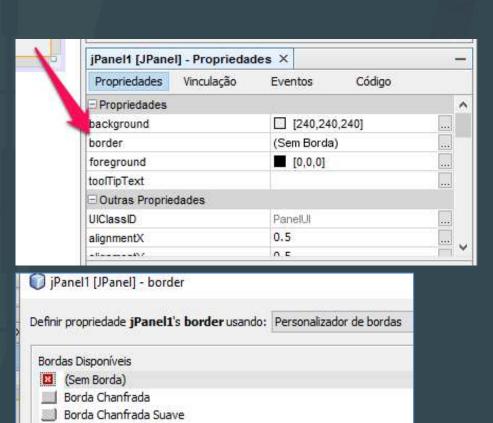
Inserindo componentes

JPanel (Painel): clicar neste componente na Paleta Swing e depois no form para adicioná-lo e redimensionar





Propriedades→ border → borda gravada



25 67

> Borda Composta Borda Fosca Borda Gravada Borda Vazia Borda com Título Borda de Linha

 Inserir os componentes Label (jLabel), Campo de Texto (jTextField), Campo de Senha (jPasswordField) e botões (jButton)

jLabel 1	jTextField1		
jLabel2	jTextField2		
jLabel3	jTextField3]	
jLabel4	•••••]	
		jButton1	jButton2

Aplicação

Login

Editar os textos em Propriedades > text e nome em Código > Nome da variável

action packground		(nenhum>	*				
ackground	r				☐ Geração de Códige		
background [240,240,240] font Tahoma 11 Simples				Classe Bean	class javax.swing.JTextField		
				Nome da Variável			
oreground		[0,0,0]	3770		Modificadores de Variáveis	private	
con		<nenhum></nenhum>	¥		Parâmetros de Tipo	• • • • • • • • • • • • • • • • • • • •	
ext	i	Button1			Utilizar Variável Local		
ORE				<u> </u>	Código de Criação Personalizado		
Button1 [JButton]			1	9	Código de Pré-Criação		
Código-Fonte	Projeto	Histórico		語 [5]			

- Inserir código para eventos:
 - clicar com o
 botão direito do
 mouse no botão
 Cancelar, ir em
 Eventos → Mouse
 → mousePressed



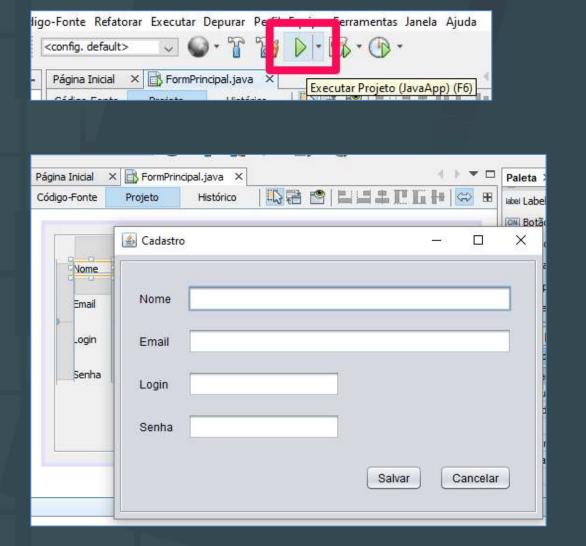
5	Vincular		Propriedades Vinculação
Cancelar	Vincular Eventos Alinhar Ancorar Redimensionamento Automático Mesmo Tamanho Definir para o Tamanho Default Incluir em Editar Espaço do Layout Criar Pai Mover para Cima Mover para Baixo	Action Ancestor Change Component Container Focus Hierarchy HierarchyBounds InputMethod Item Key Mouse	> Eventos ctionPerformed incestorAdded incestorMoved incestorMoved incestorRemoved incestorMoved incestorMoved incestorMoved incestorMoved incestorMoved incestorMoved incestorMoved incestorMoved incestorRemoved incestorRemoved
	Recortar	MouseWheel	mouseEntered mouseExited
	Copiar Duplicar Excluir	PropertyChange VetoableChange	mousePressed mouseReleased

```
Página Inicial X
             FormPrincipal.java X
                                 Código-Fonte
 134
                   .addGroup(layout.createSequentialGroup()
 135
                       .addContainerGap()
                       .addComponent(jPanell, javax.swing.GroupLayout.PREFERRED SIZE, javax
 136
 137
                       .addContainerGap(javax.swing.GroupLayout.DEFAULT SIZE, Short.MAX VA
 138
               1:
 139
 140
               pack();
           }// </editor-fold>
 141
 142
 143
           private void jbtCancelarMousePressed(java.awt.event.MouseEvent evt) {
               // TODO add your handling code here:
 144
 145
 146
 147
            * Sparam args the command line arguments
 148
```

```
private void jbtCancelarMousePressed(java.awt.event.MouseEvent evt) {
    System.exit(0); // TODO add your handling code here:
}
```

Para rodar a aplicação, clicar no botão Executar Projeto. Uma caixa se abrirá para seleção da classe principal. Clicar em OK





Tema 4 A Biblioteca Gráfica OpenGL em Linguagem C

Graphics Library) é
uma API
(Application
Programming
Interface) criada
em 1992

OpenGL (Open

Características

- Objetos 2D/3D
- Primitivasgeométricas
- Imagens vetoriais e matriciais

- Sistema de janelas independente
- Suporte à iluminação, sombra, textura, transparência, animação, eventos e transformações

Medidas de ângulos em graus

Representação de cor com valor fracionários de 0 a 1

Coordenadascartesianas2D ou 3D

- Tipos de dados próprios
- Opera como uma máquina de estados



Primitivas geométricas

Para a criação, é necessário delimitá-las pelos comandos glBegin e glEnd Ponto (vértice)
Primitiva GL_POINTS

```
Exemplo 2D:

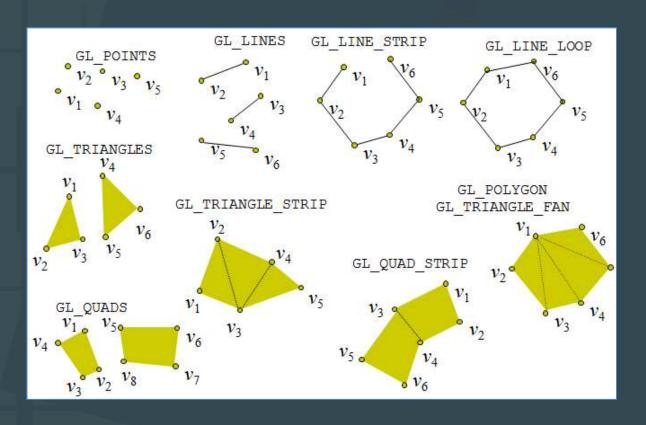
glBegin( GL_POINTS );

glVertex2f( xf, yf);

glEnd();
```

```
Linha
Primitivas:
GL_LINES (2 vértices)
GL_LINE_STRIP (linhas consecutivas)
GL_LINE_LOOP (último vértice ligado ao primeiro)
Exemplo 2D:
glBegin(GL_LINES);
  glVertex2i(40, 100);
  glVertex2i(500, 400);
glEnd();
```

Primitivas Geométricas



```
Exemplo com GL_POLYGON:
glBegin(GL_POLYGON);
   glVertex3f (1, 2, 0.0);
   glVertex3f (2, 2, 0.0);
   glVertex3f (2, 3, 0.0);
   glVertex3f (1, 3, 0.0);
 glEnd();
```

Transformações geométricas

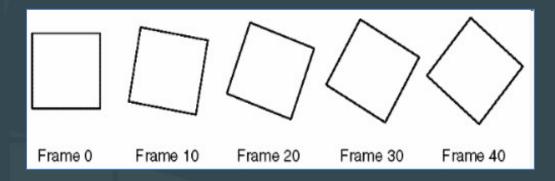
Sequência armazenada em uma pilha, por meio das funções glPushMatrix e glPopMatrix

- Translação (glTranslatef)
 - Move a origem do sistema
- Escala (glScalef)
 - Amplia o sistema de coordenadas
- Rotação (glRotatef)
 - Em graus no sentido anti-horário

Animação

- Usa buffer duplo, uma imagem sendo mostrada na tela e outra sendo processada para ocupar o lugar dela
- Usa função glutSwapBuffers

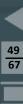
 O processamento deve levar em conta a taxa de atualização



Cores

- Formato de cores RGBA (red, green, blue, alpha)
- Preenchimento:
 - Sombra (shading)
 - Cor cheia (flat)
 - Cores variadas (smooth)
- Transparência:
 - Habilitar opção blend







Iluminação

- Formas:
 - Ambiente: luz dispersa igualmente em todas as direções
 - Difusa: vem de uma direção

Especular: é como uma luz refletida em um espelho, um brilho

- Reflexão do objeto
 - Ambiente e difusa: similares à cor do objeto
 - Especular: geralmente branca ou cinza



Passos para adicionar luz à cena

Definir o vetor normal de cada vértice (para onde aponta) de todos os objetos, para determinar a orientação do objeto em relação à fonte de luz

- Criar, selecionar e posicionar uma ou mais fontes de luz
- Criar e selecionar um modelo de iluminação
- Definir as propriedades de material do objeto da cena

Tema 5 Projetos Utilizando OpenGL para Animações Gráficas

OpenGL com Visual Studio

Projeto Win32
Console
Application

Manage Packages for Solution	
Browse Installed Updates Consolidate	Package source: nuget.org - 🗘
nupengl X 🔻 🗌 Include prerelease	
nupengl.core.redist by Jonathan Dickinson, Ali Badereddin, 1 Redistributable components for for package 'nupengl.core'	version(s) - 1
nupengl.core by Jonathan Dickinson, Ali Badereddin, 12,1K do	✓ Project ^ Version
NupenGL allows you to access OpenGL from your application.	✓ OpenGL 0.1.0.1
	Installed: 0.1.0.1 Uninstall
	Version: Latest stable 0.1. Install

- Exemplo: quadrado animado rotacionando continuamente
 - Primitivas geométricas
 - Transformações geométricas
 - Animação
 - Interação com o mouse

- Funções do código:
 - main
 - display
 - spinDisplay
 - reshape
 - Mouse

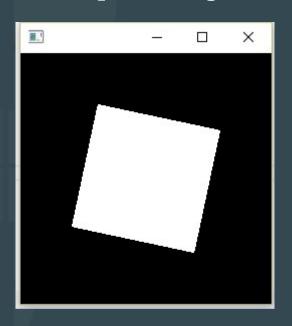
```
main:
static GLfloat spin = 0.0; // variável global para o ângulo
int main(int argc, char** argv)
      glutInit(&argc, argv);
      // sistema de cores para RGB e buferização dupla
      glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
      glutInitWindowSize(250, 250); // tamanho da janela
      glutInitWindowPosition(100, 100);
      glutCreateWindow(argv[0]);
      // cor de fundo
      glClearColor(0.0, 0.0, 0.0, 0.0);
      glShadeModel(GL_FLAT);
      // Callbacks
      glutDisplayFunc(display);
      glutReshapeFunc(reshape);
      glutMouseFunc(mouse);
      glutMainLoop();
      return 0;
```

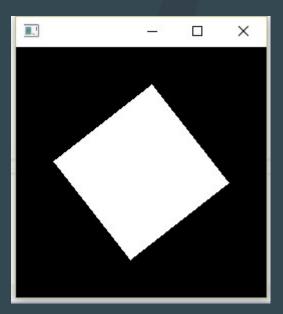
```
display:
// desenha o quadrado branco
void display(void)
       // fundo preto
       glClear(GL_COLOR_BUFFER_BIT);
       glPushMatrix();
        //rotação no eixo x conforme variável spin
       glRotatef(spin, 0.0, 0.0, 1.0);
        // cor branca
       glColor3f(1.0, 1.0, 1.0);
       // retângulo
       glRectf(-25.0, -25.0, 25.0, 25.0);
       glPopMatrix();
       // animação
       glutSwapBuffers();
```

```
reshape:
   // trata o redimensionamento da janela
  void reshape(int w, int h)
          glViewport(0, 0, (GLsizei)w, (GLsizei)h);
          glMatrixMode(GL_PROJECTION);
          glLoadIdentity();
          glOrtho(-50.0, 50.0, -50.0, 50.0, -1.0, 1.0);
          glMatrixMode(GL_MODELVIEW);
          glLoadIdentity();
62
67
  // ativa a animação
  void spinDisplay(void)
          spin = spin + 2.0;
          if (spin > 360.0)
                 spin = spin - 360.0;
          glutPostRedisplay();
```

```
mouse:
// trata os cliques do mouse
void mouse(int button, int state, int x, int y)
      switch (button) {
      // botão esquerdo ativa a animação
      case GLUT_LEFT_BUTTON:
            if (state == GLUT_DOWN)
                  glutIdleFunc(spinDisplay);
            break;
      // botão direito desativa
      case GLUT_RIGHT_BUTTON:
            if (state == GLUT_DOWN)
                  glutIdleFunc(NULL);
            break;
      default:
            break;
```

Aplicação rodando:





 Dois momentos da rotação do quadrado, que permanece continuamente rotacionando.

Síntese

- Nesta terceira aula vimos:
 - A linguagem Java
 - A IDE NetBeans
 - Interface gráfica Java
 - A biblioteca
 OpenGL
 - AnimaçãoOpenGL