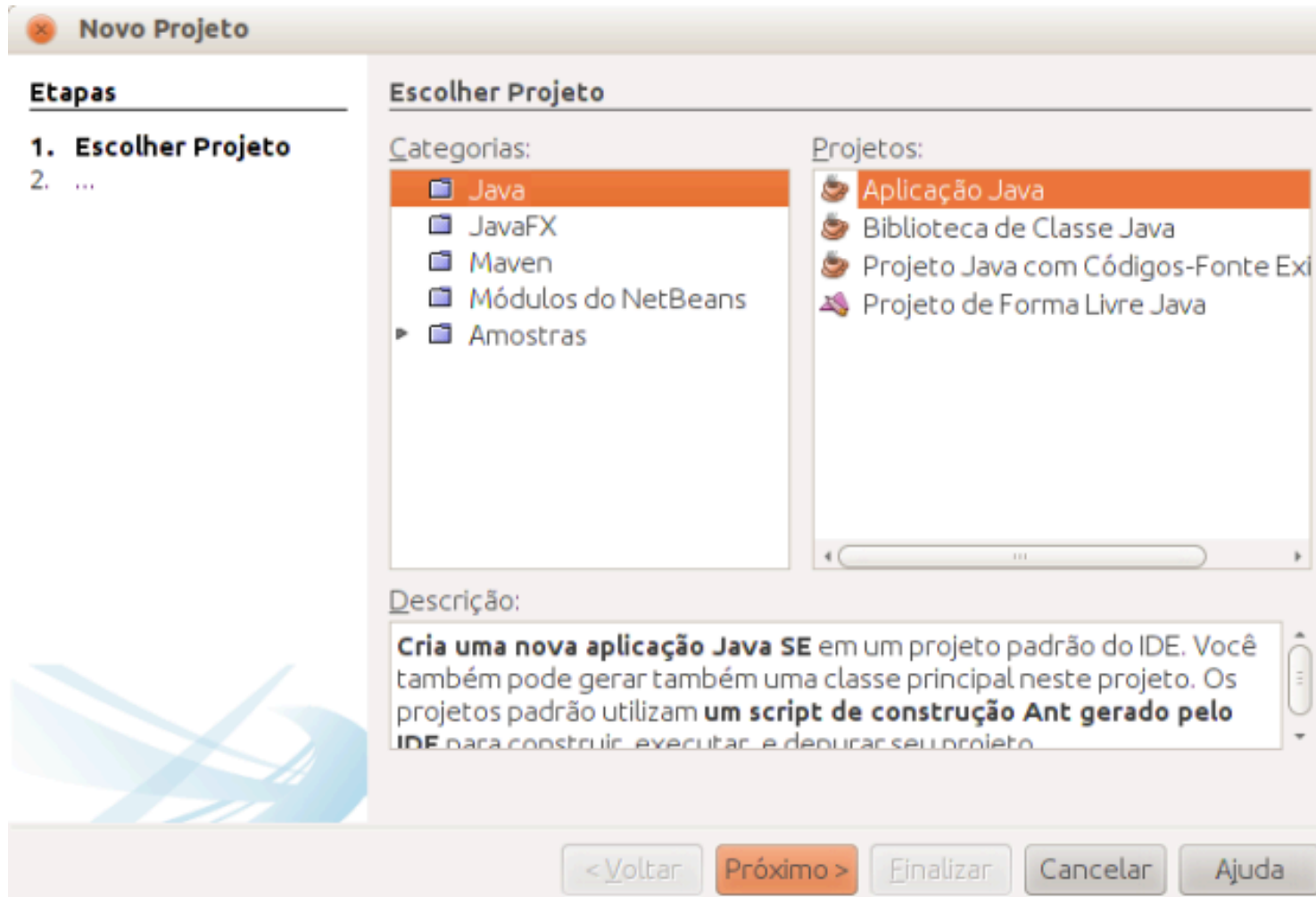
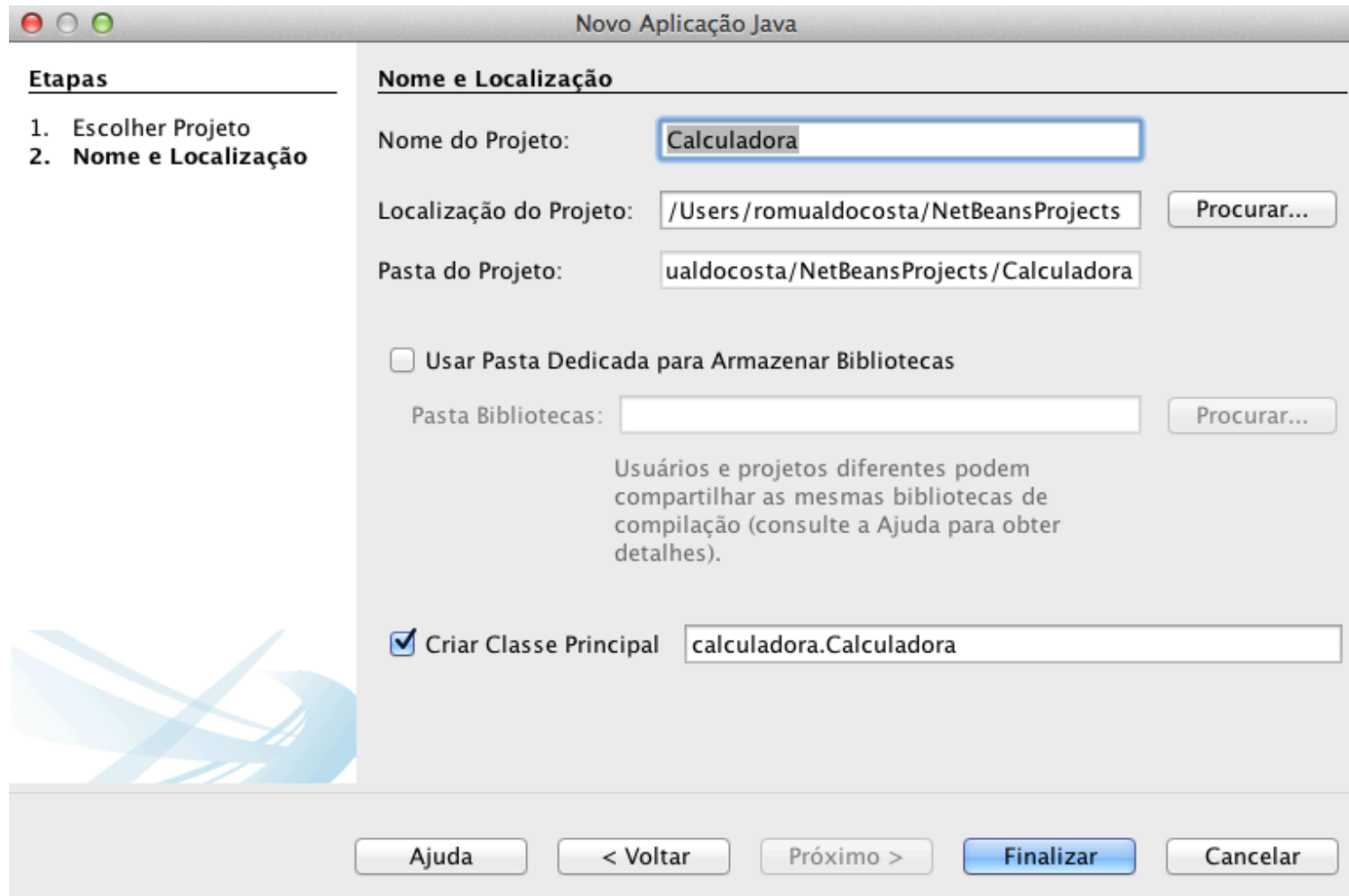


NetBeans – Novo Projeto



NetBeans



The image shows the 'Novo Aplicação Java' (New Java Application) dialog box in NetBeans. The window has a title bar with standard Mac OS window controls (red, yellow, green buttons) and the title 'Novo Aplicação Java'. On the left, there is a sidebar titled 'Etapas' (Steps) with two items: '1. Escolher Projeto' and '2. Nome e Localização', where the second item is currently selected. The main area is titled 'Nome e Localização' (Name and Location). It contains several input fields and buttons. The 'Nome do Projeto' (Project Name) field is filled with 'Calculadora'. The 'Localização do Projeto' (Project Location) field is filled with '/Users/romualdocosta/NetBeansProjects', with a 'Procurar...' (Browse...) button to its right. The 'Pasta do Projeto' (Project Folder) field is filled with 'ualdocosta/NetBeansProjects/Calculadora'. Below these, there is an unchecked checkbox labeled 'Usar Pasta Dedicada para Armazenar Bibliotecas' (Use Dedicated Folder for Storing Libraries). Below this checkbox is a 'Pasta Bibliotecas' (Library Folder) field and another 'Procurar...' button. A paragraph of text explains that different users and projects can share the same libraries. At the bottom of the main area, there is a checked checkbox labeled 'Criar Classe Principal' (Create Main Class), followed by a text field containing 'calculadora.Calculadora'. At the very bottom of the dialog, there are five buttons: 'Ajuda' (Help), '< Voltar' (Back), 'Próximo >' (Next), 'Finalizar' (Finish), and 'Cancelar' (Cancel). The 'Finalizar' button is highlighted in blue.

Etapas

1. Escolher Projeto
2. **Nome e Localização**

Nome e Localização

Nome do Projeto:

Localização do Projeto:

Pasta do Projeto:

☐ Usar Pasta Dedicada para Armazenar Bibliotecas

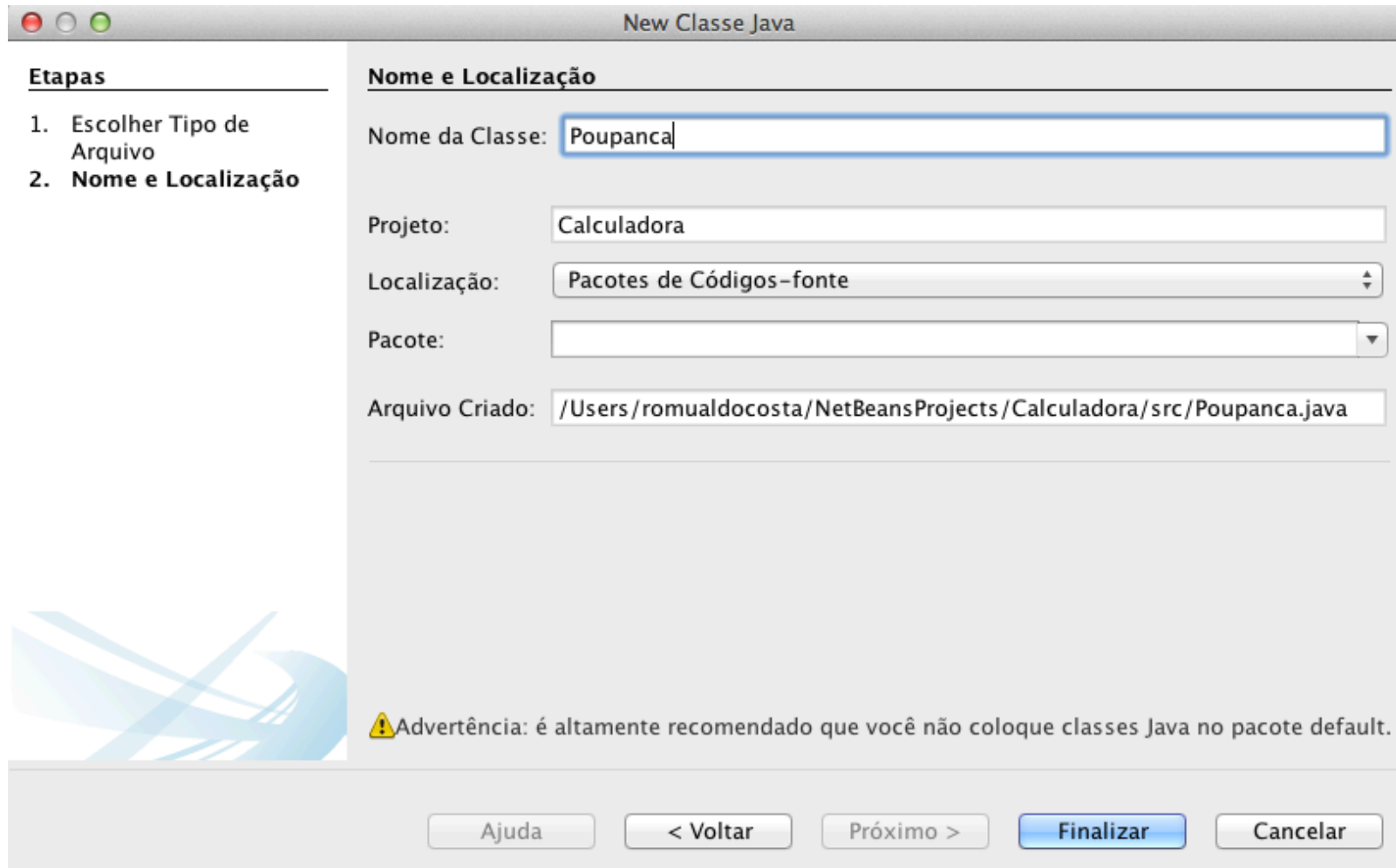
Pasta Bibliotecas:

Usuários e projetos diferentes podem compartilhar as mesmas bibliotecas de compilação (consulte a Ajuda para obter detalhes).

☒ Criar Classe Principal



NetBeans – Nova Classe



The image shows the 'New Classe Java' dialog box in the NetBeans IDE. The window has a title bar with standard Mac OS window controls (red, yellow, green buttons) and the title 'New Classe Java'. On the left, there is a sidebar titled 'Etapas' (Steps) with two items: '1. Escolher Tipo de Arquivo' and '2. Nome e Localização', where the second item is currently selected. The main area is titled 'Nome e Localização' (Name and Location) and contains several input fields: 'Nome da Classe:' with the text 'Poupanca' entered; 'Projeto:' with 'Calculadora' entered; 'Localização:' with a dropdown menu showing 'Pacotes de Códigos-fonte'; 'Pacote:' with an empty dropdown menu; and 'Arquivo Criado:' with the path '/Users/romualdocosta/NetBeansProjects/Calculadora/src/Poupanca.java' entered. At the bottom of the main area, there is a warning icon and the text: 'Advertência: é altamente recomendado que você não coloque classes Java no pacote default.' The bottom of the dialog features five buttons: 'Ajuda', '< Voltar', 'Próximo >', 'Finalizar' (highlighted in blue), and 'Cancelar'.

Etapas

1. Escolher Tipo de Arquivo
2. Nome e Localização

Nome e Localização

Nome da Classe: Poupanca

Projeto: Calculadora

Localização: Pacotes de Códigos-fonte

Pacote:

Arquivo Criado: /Users/romualdocosta/NetBeansProjects/Calculadora/src/Poupanca.java

⚠ Advertência: é altamente recomendado que você não coloque classes Java no pacote default.

Ajuda < Voltar Próximo > Finalizar Cancelar



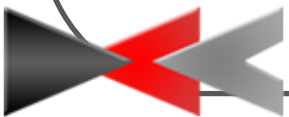
```
public class Poupanca {  
    private int anos;  
    private double juros;  
    private double depositoMensal;  
  
    public Poupanca(int a, double j, double dm) {  
        anos = a;  
        juros = j/100;  
        depositoMensal = dm;  
    }  
  
    public double calculaTotal() {  
        int num_pagamentos = anos * 12;  
        double total = 0;  
        for (int i = 0; i < num_pagamentos; i++) {  
            total = total + depositoMensal;  
            total = total + total * juros;  
        }  
        return total;  
    }  
}
```



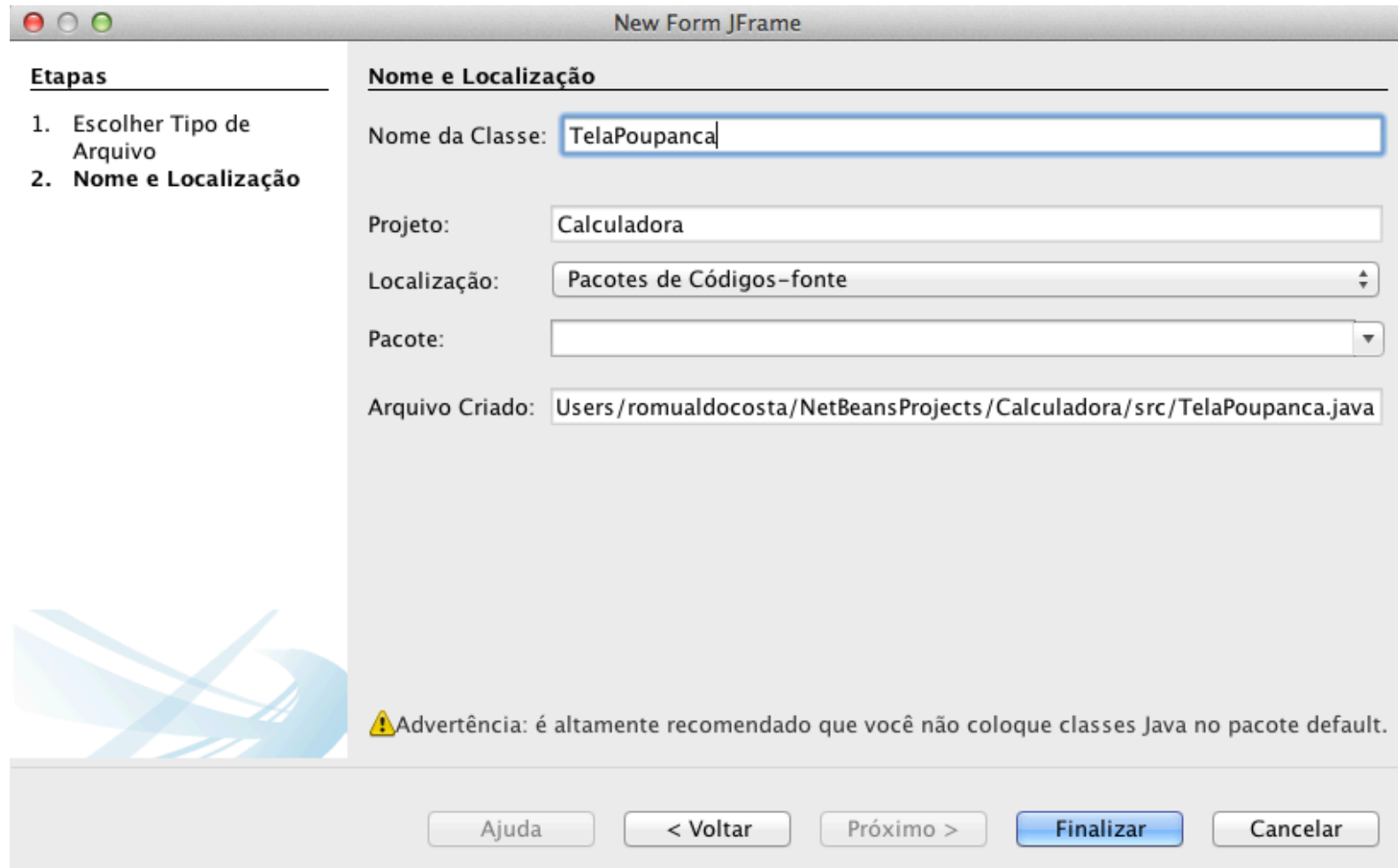
```
public void setAnos(int anos) {  
    this.anos = anos;  
}  
public void setJuros(double juros) {  
    this.juros = juros/100;  
}  
public void setDepositoMensal(double depositoMensal) {  
    this.depositoMensal = depositoMensal;  
}  
public int getAnos() {  
    return anos;  
}  
public double getJuros() {  
    return juros;  
}  
public double getDepositoMensal() {  
    return depositoMensal;  
}
```



```
public String toString() {  
    return "Anos: " + anos + "\nJuros: " + juros  
        + "\nDepósito mensal: " + depositoMensual;  
}  
}
```



NetBeans – JFrame



The image shows the 'New Form JFrame' dialog box in NetBeans. The window title is 'New Form JFrame'. On the left, under the 'Etapas' (Steps) section, there are two steps: '1. Escolher Tipo de Arquivo' and '2. Nome e Localização', with the second step being the active one. The main area is titled 'Nome e Localização' (Name and Location). It contains several fields: 'Nome da Classe:' with the text 'TelaPoupanca' entered; 'Projeto:' with 'Calculadora' entered; 'Localização:' with a dropdown menu showing 'Pacotes de Códigos-fonte'; 'Pacote:' with an empty dropdown menu; and 'Arquivo Criado:' with the path 'Users/romualdocosta/NetBeansProjects/Calculadora/src/TelaPoupanca.java' entered. At the bottom, there is a warning message: 'Advertência: é altamente recomendado que você não coloque classes Java no pacote default.' Below the warning are five buttons: 'Ajuda', '< Voltar', 'Próximo >', 'Finalizar' (highlighted in blue), and 'Cancelar'.

Etapas

1. Escolher Tipo de Arquivo
2. Nome e Localização

Nome e Localização

Nome da Classe:

Projeto:

Localização:

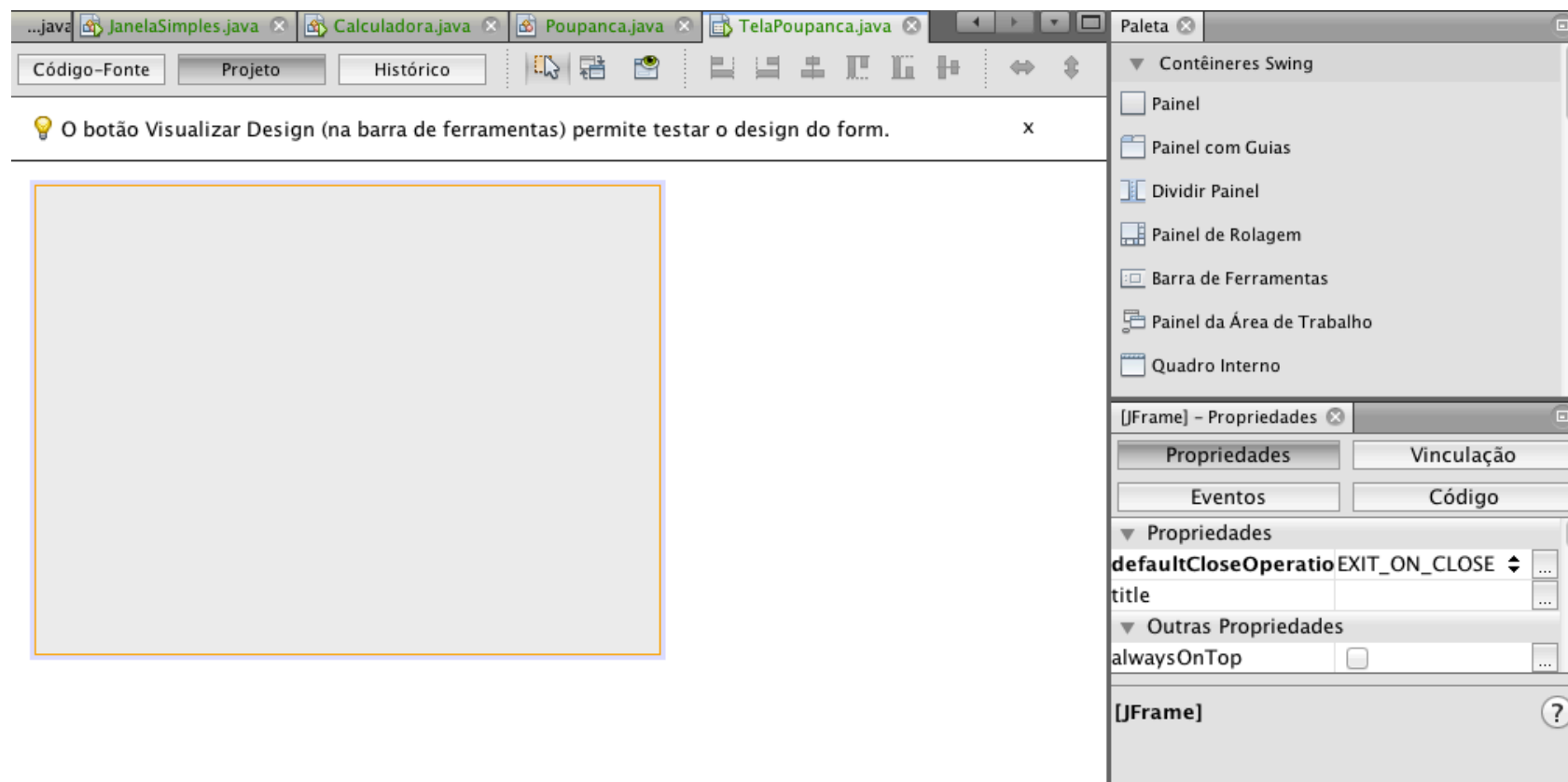
Pacote:

Arquivo Criado:

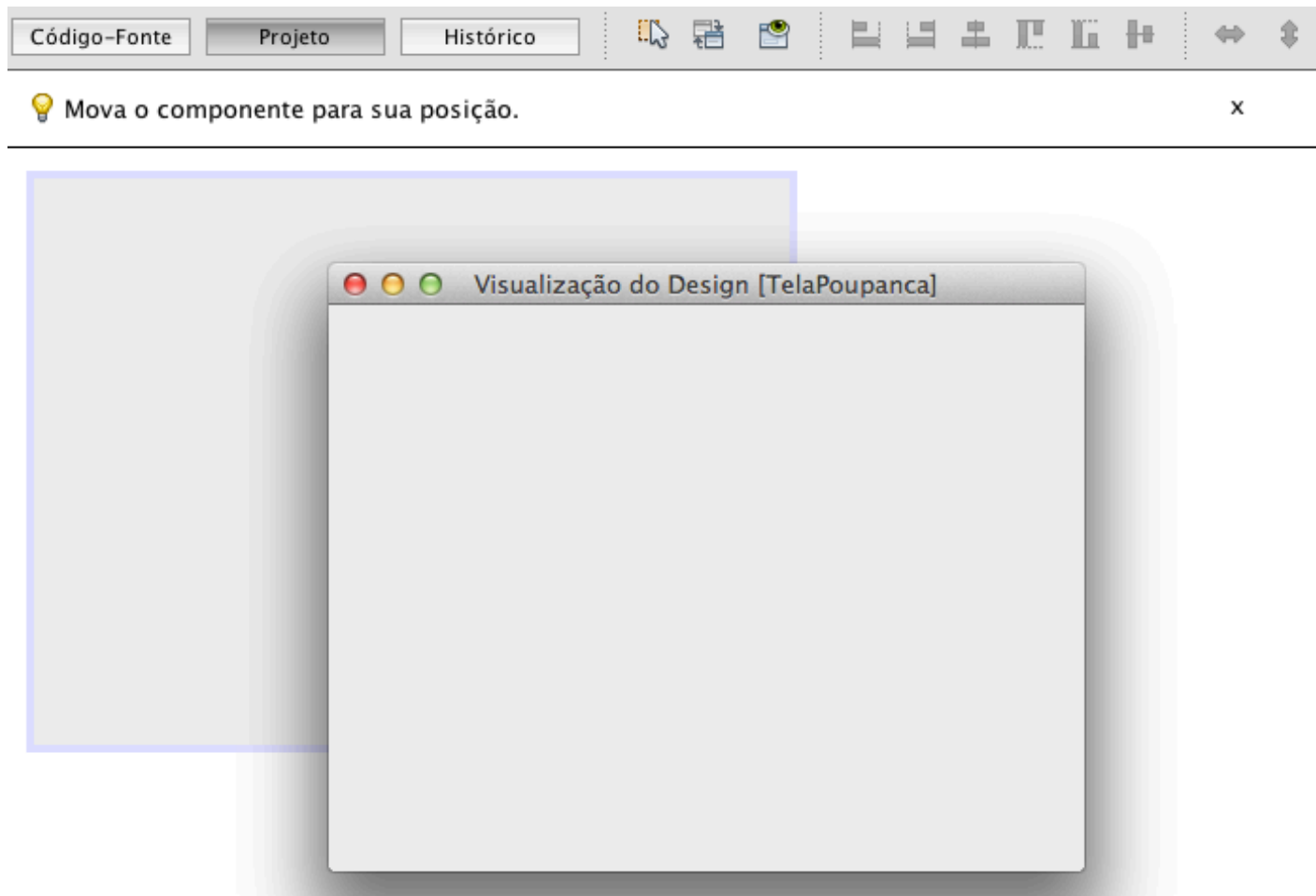
⚠ Advertência: é altamente recomendado que você não coloque classes Java no pacote default.



NetBeans – Área de Design

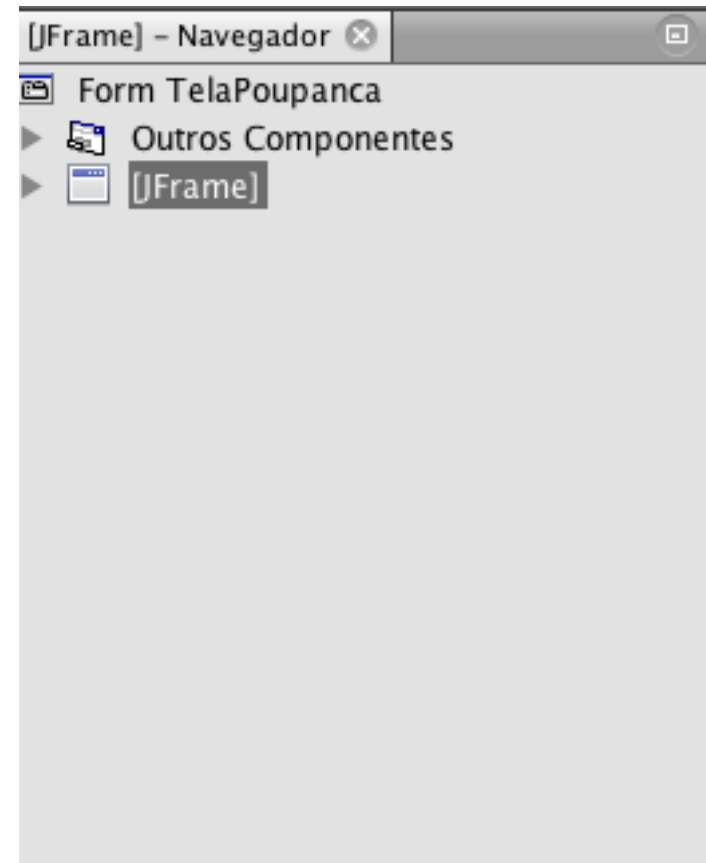


NetBeans – Visualização



NetBeans – Navegador

- Fornece uma representação de todos os componentes, em uma hierarquia de árvore



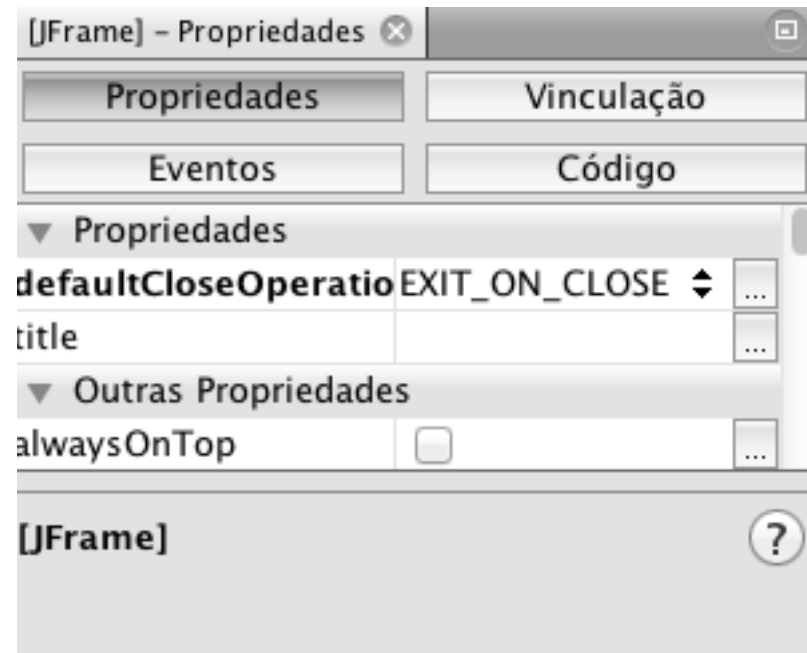
NetBeans – Navegador

- Paleta: Lista de componentes disponíveis contendo guias jfc/swing e awt bem como gerentes de layout

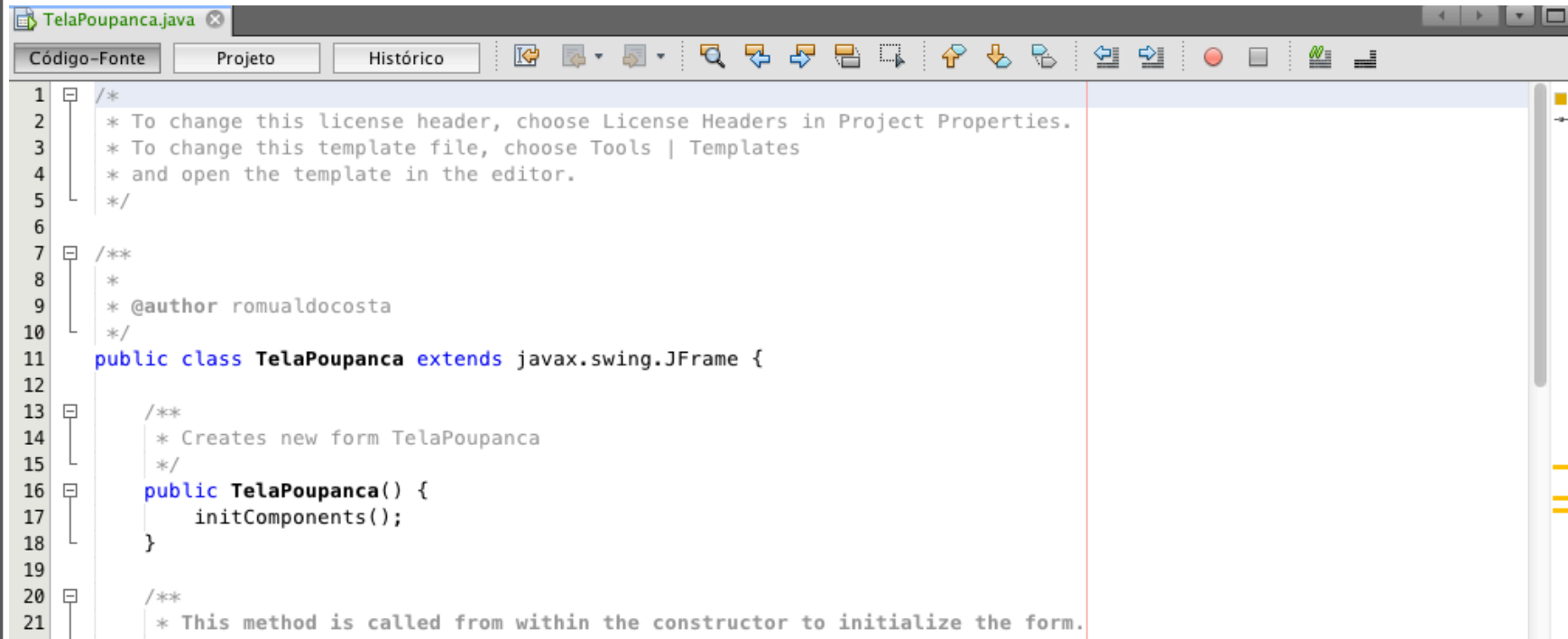


NetBeans – Navegador

- Propriedades: exibe as propriedades do componente atualmente selecionado no navegador GU Builder



NetBeans – Código Fonte



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6
7  /**
8   *
9   * @author romualdocosta
10  */
11  public class TelaPoupanca extends javax.swing.JFrame {
12
13      /**
14       * Creates new form TelaPoupanca
15       */
16      public TelaPoupanca() {
17          initComponents();
18      }
19
20      /**
21       * This method is called from within the constructor to initialize the form.
```

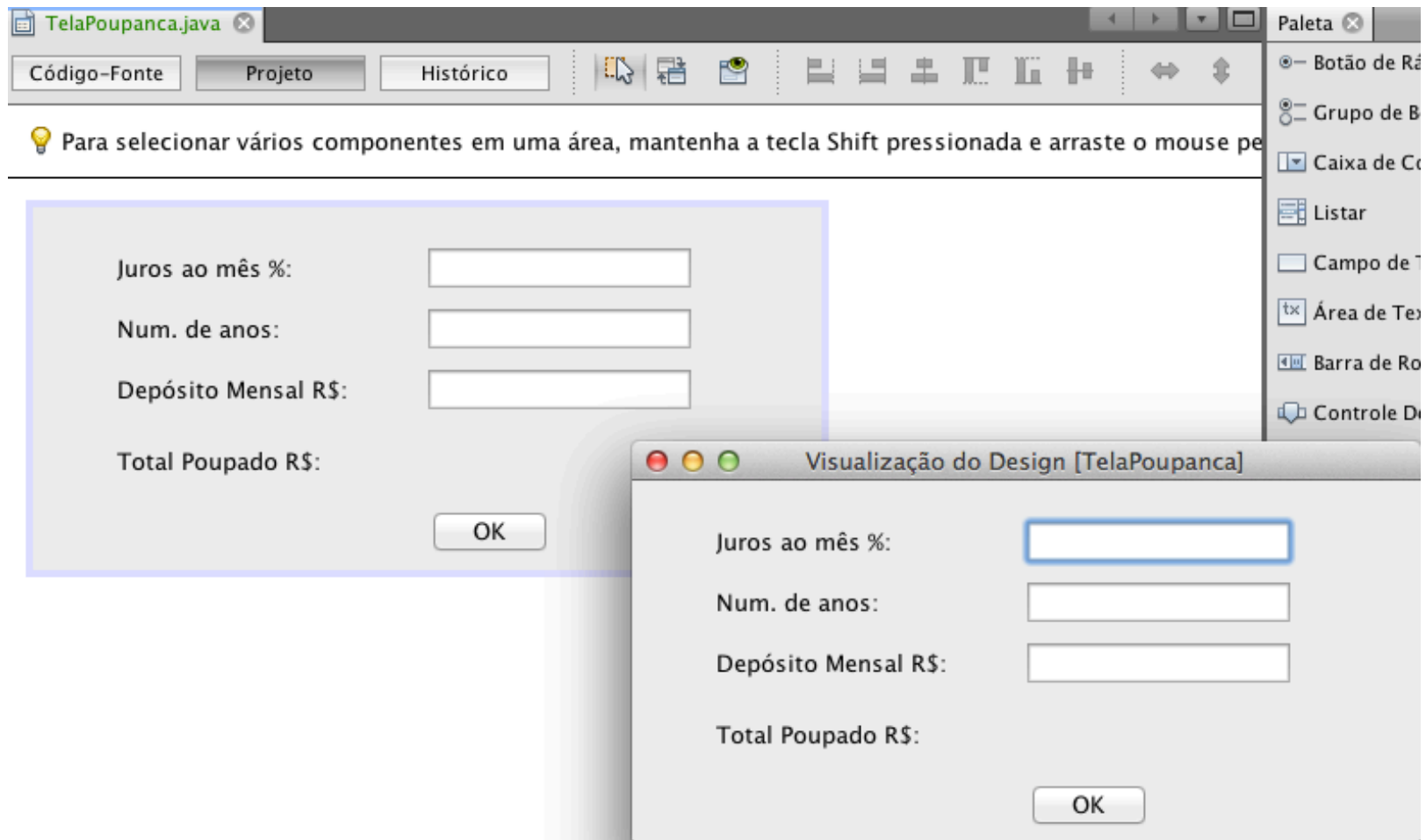


GUI

- Na paleta, arraste o painel até o formulário do projeto
- Arraste os labels, campos de texto e botões e altere os nomes e seus valores
- Inserção do evento de seleção do botão



Tela Poupança



Eventos



Orientação por Eventos

- Um modelo de programação que tornou-se bastante difundido com o uso de interfaces gráficas foi a programação orientada a eventos.
- Nesse modelo o programação deixa de ter o controle do fluxo de execução que passa a um sistema encarregado de gerenciar a interface
- O programa passa a ser chamado quando algum evento é gerado na interface



Callback

- Para que o programa possa ser chamado pelo sistema, ele deve registrar uma função para cada evento de interface que ele desejar tratar
- Cada função é chamada de callback (chamadas de volta) do sistema



Callback OO

- No sistema OO o modelo de orientação por eventos deve implementar um objeto
 - Não possui o conceito de função
- Onde seria registrada uma função passa a ser registrado um objeto
 - A execução a um callback passa por um método de um objeto onde o evento é tratado.



Callback Java

- Um objeto que implementa uma callback em Java é chamado listener.
- Listener são definidos por interfaces e podem estar aptos a tratar mais de um tipo de evento.
- Quando um listener tem um de seus métodos chamados, ele recebe um parâmetro que descreve o evento ocorrido.
- Esse parâmetro é um objeto. Existem classes para diferentes tipos de eventos.



Listeners

- Definem interfaces que representam um grupo de callbacks
- São registrados junto aos componentes
- Exemplo: `java.awt.event.MouseListener`
 - `public abstract void mouseClicked(MouseEvent e)`
 - `public abstract void mousePressed(MouseEvent e)`
 - `public abstract void mouseReleased(MouseEvent e)`
 - `public abstract void mouseEntered(MouseEvent e)`
 - `public abstract void mouseExited(MouseEvent e)`

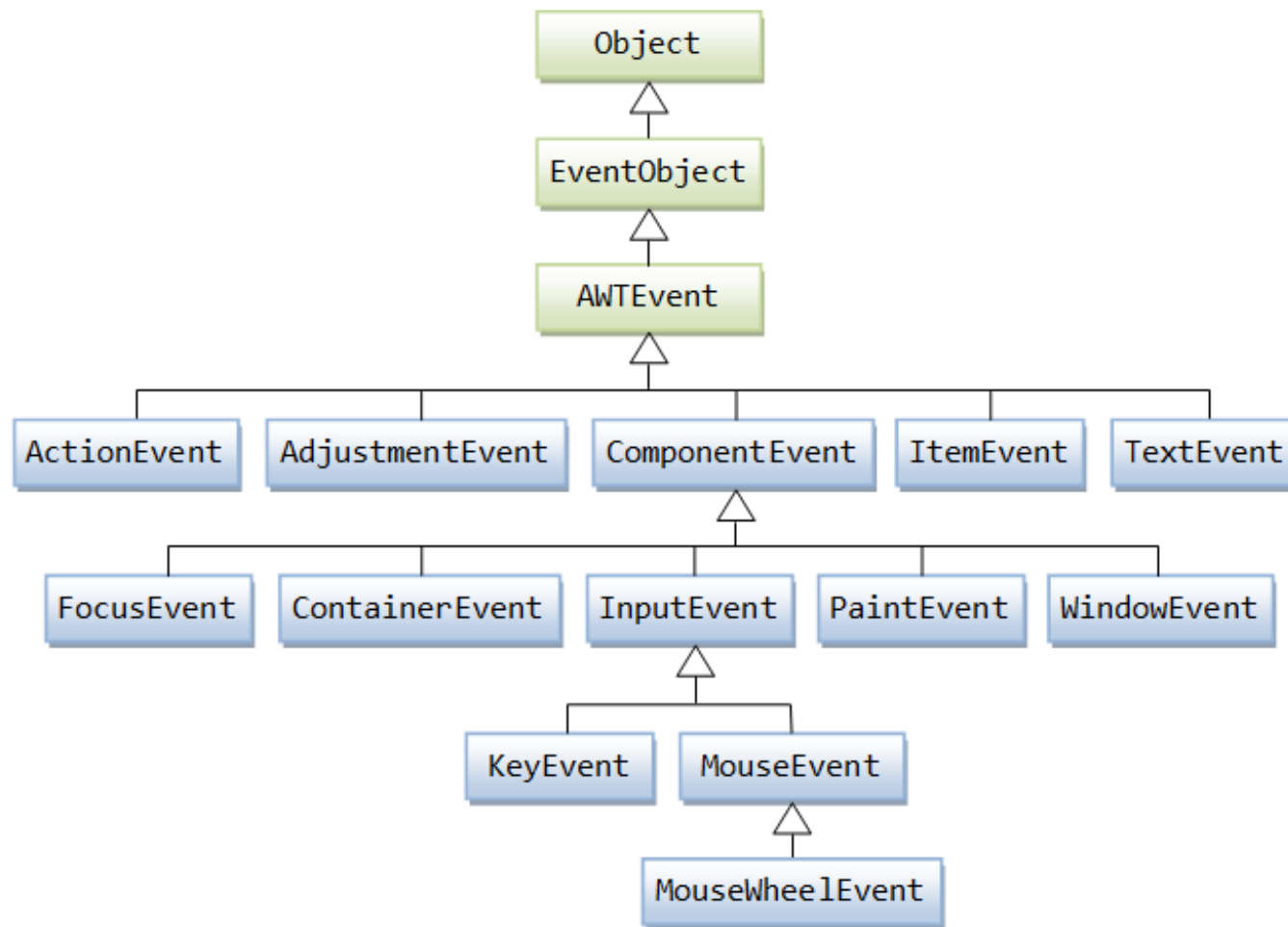


Eventos

- Trazem informações sobre o evento ocorrido
- São passados aos listeners (callbacks)
- Exemplo: `java.awt.event.MouseEvent`
 - `public int getX()`
 - `public int getY()`
 - `public int getClickCount()`



Alguns Eventos



Eventos

- Trazem informações sobre o evento ocorrido
- São passados aos listeners (callbacks)
- Exemplo: `java.awt.event.MouseEvent`
 - `public int getX()`
 - `public int getY()`
 - `public int getClickCount()`



Implementando um Listener

- Defina a classe que implementa a interface do Listener

```
class MeuListener implements ActionListener {  
    public void actionPerformed(ActionEvent e) {  
        System.out.println("Botão pressionado");  
    }  
}
```



Registrando um Listener

- Crie o objeto callback e o registre no componente

```
button.addActionListener(new MeuListener());
```



Tratando Eventos no NetBeans

- Escolha o componente, clique no botão direito escolha a opção Events.
- Exemplo
 - Inserir um botão para sair
 - Escolha Action=>actionPerformed
 - Dentro do método escreva `System.exit(0)`



Ou, no exemplo da calculadora

- Escolha o componente, clique no botão direito escolha a opção Events.
 - Escolha Action=>actionPerformed
 - E escreva o código referente ao cálculo:
- Mas é MouseClicked ou ActionPerformed:
 - ActionPerformed é a ação padrão do componente, no caso do botão, ele pode ser acionado pelo teclado.



Form layout with labels and input fields:

- Juros ao mês %:
- Num. de anos:
- Depósito Mensal R\$:
- Total Poupado R\$:

- Editar Texto
- Alterar o Nome da Variável...
- 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
- Recortar
- Copiar
- Duplicar
- Deletar
- Personalizar Código...
- Propriedades

- Action
- Ancestor
- Change
- Component
- Container
- Focus
- Hierarchy
- HierarchyBounds
- InputMethod
- Item
- Key
- Mouse
- MouseMotion
- MouseWheel
- PropertyChange
- VetoableChange



```
private void okActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    int anos, deposito;  
    double juros;  
    try {  
        anos = Integer.parseInt(txtAnos.getText());  
        juros = Double.parseDouble(txtJuros.getText());  
        deposito = Integer.parseInt(txtDeposito.getText());  
    } catch (NumberFormatException e) {  
        anos = 0;  
        juros = 0;  
        deposito = 0;  
    }  
    poupanca.setAnos(anos);  
    poupanca.setDepositoMensal(deposito);  
    poupanca.setJuros(juros);  
  
    lblResultado.setText(String.valueOf((poupanca.calculaTotal())));  
}
```



```
public class TelaPoupanca extends javax.swing.JFrame {  
    Poupanca poupanca;  
  
    public void setPoupanca(Poupanca poupanca) {  
        this.poupanca = poupanca;  
    }  
}
```



```
public class Calculadora {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        Poupanca poupanca = new Poupanca(0,0,0);  
        TelaPoupanca tela = new TelaPoupanca();  
        tela.setPoupanca(poupanca);  
        tela.setVisible(true);  
    }  
}
```



Exercício

- Calculadora:
 - Com visor
 - Botão CE
 - Botões de números
 - Operações $+$, $-$, $/$ e $*$



Dicas

- Faça o visor com um painel de texto:
 - O evento de cada botão numérico corresponde a pegar o conteúdo do painel de texto e adicionar o seu valor.
 - Quando o operador for pressionado o número é colocado em uma pilha junto com o operador
 - O visor é limpo para novas adições
 - Quando o operador = é pressionado a pilha é calculada.

