

# Ferramenta para a Construção de Hierarquia de Classes em Ambiente

Java



José Raphael de Souza Lamas

Arthur de Almeida Zambelli

Orientadora: Mariza Bigonha



# Tópicos Abordados

- Introdução
- Class Designer Java
- Telas do Class Designer Java
- Conclusão



# Introdução

- Orientação por objetos
- O que é Class Designer ?



# Class Designer Java

## ■ Características

- Tipos de arquivos
- Primeiro passo-> definir conjunto de classes
- Membros podem ser definidos posteriormente
- Operações sobre membros e classes
  - Cópia, Movimentação e Remoção



# Class Designer Java

## ■ Características

- Classes podem ser criadas por generalização
- Pesquisa de componentes
- Pré compilação dos cabeçalhos das classes e seus membros

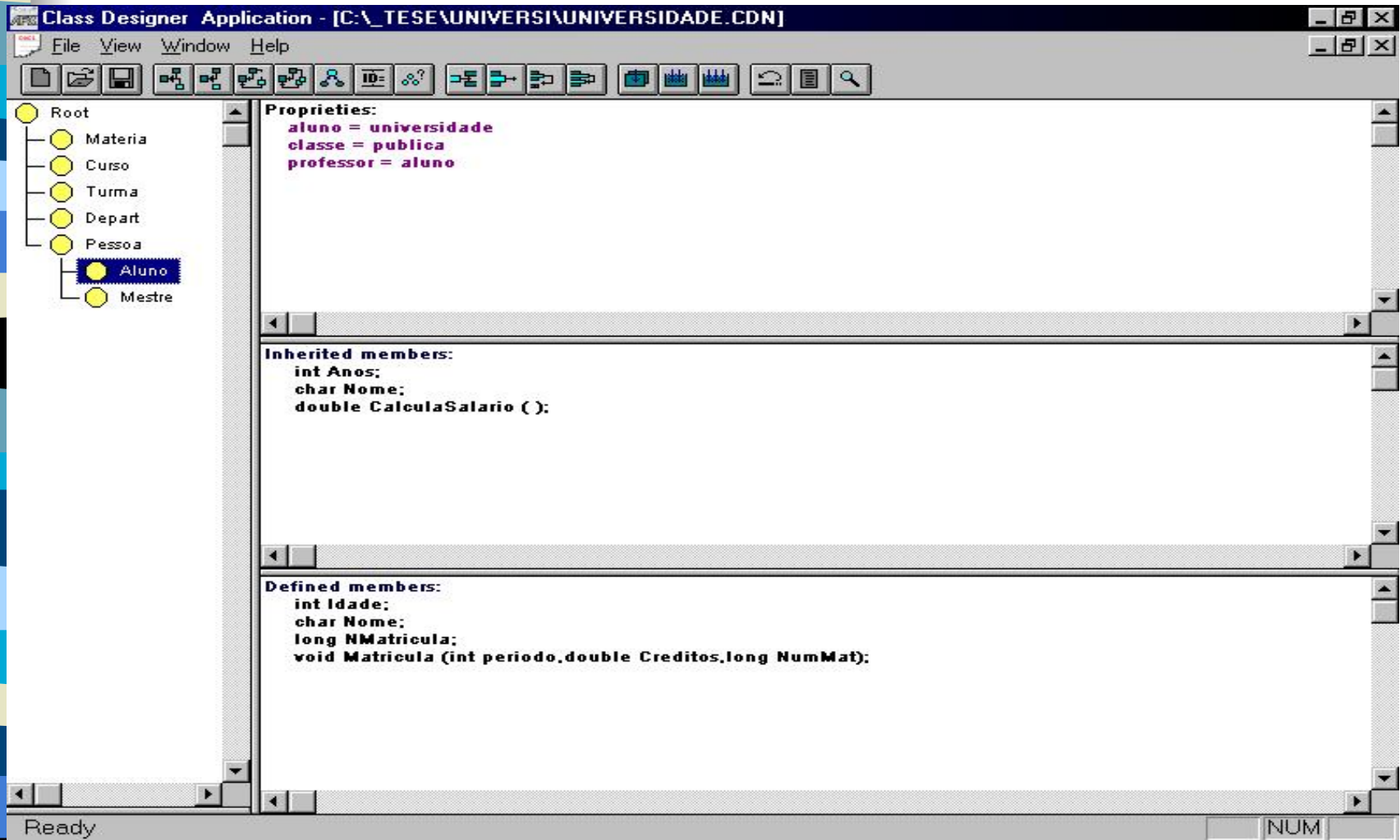


# Class Designer Java

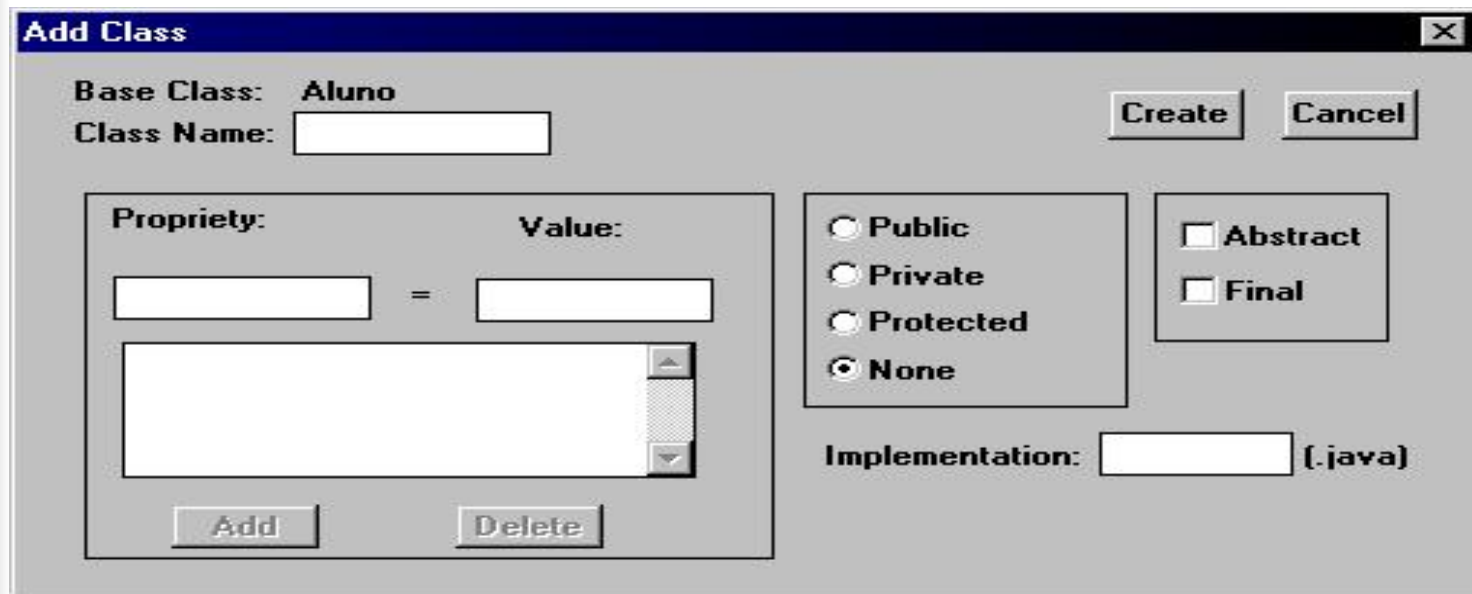
## ■ Vantagens:

- geração de arquivos que serão mantidos automaticamente
- localização e recuperação de classes por meio de assinatura ou propriedades
- geração da hierarquia a partir de arquivos fonte de Java

# Tela Principal - Class Designer



# Inserção de Classe



The image shows a 'Add Class' dialog box with a dark blue title bar and a close button. The dialog is divided into several sections. At the top left, 'Base Class:' is set to 'Aluno', and 'Class Name:' is followed by an empty text box. To the right are 'Create' and 'Cancel' buttons. Below this is a table with two columns: 'Propriety:' and 'Value:'. The first row has empty text boxes for both, separated by an equals sign. Below the table is a large empty text area with scrollbars, and 'Add' and 'Delete' buttons. To the right of the table are radio buttons for 'Public', 'Private', 'Protected', and 'None' (which is selected). Further right are checkboxes for 'Abstract' and 'Final'. At the bottom right, 'Implementation:' is followed by an empty text box and the file extension '(.java)'.

**Add Class**

Base Class: **Aluno**

Class Name:

**Create** **Cancel**

Propriety:		Value:
<input type="text"/>	=	<input type="text"/>
<input type="text"/>		

**Add** **Delete**

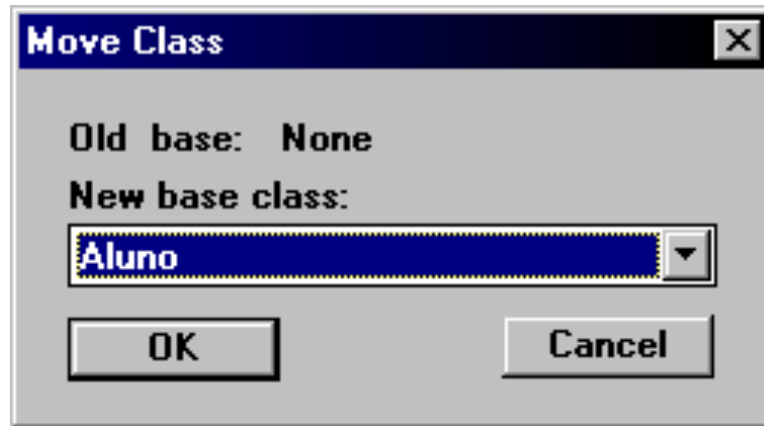
☐ **Public**  
☐ **Private**  
☐ **Protected**  
☒ **None**

☐ **Abstract**  
☐ **Final**

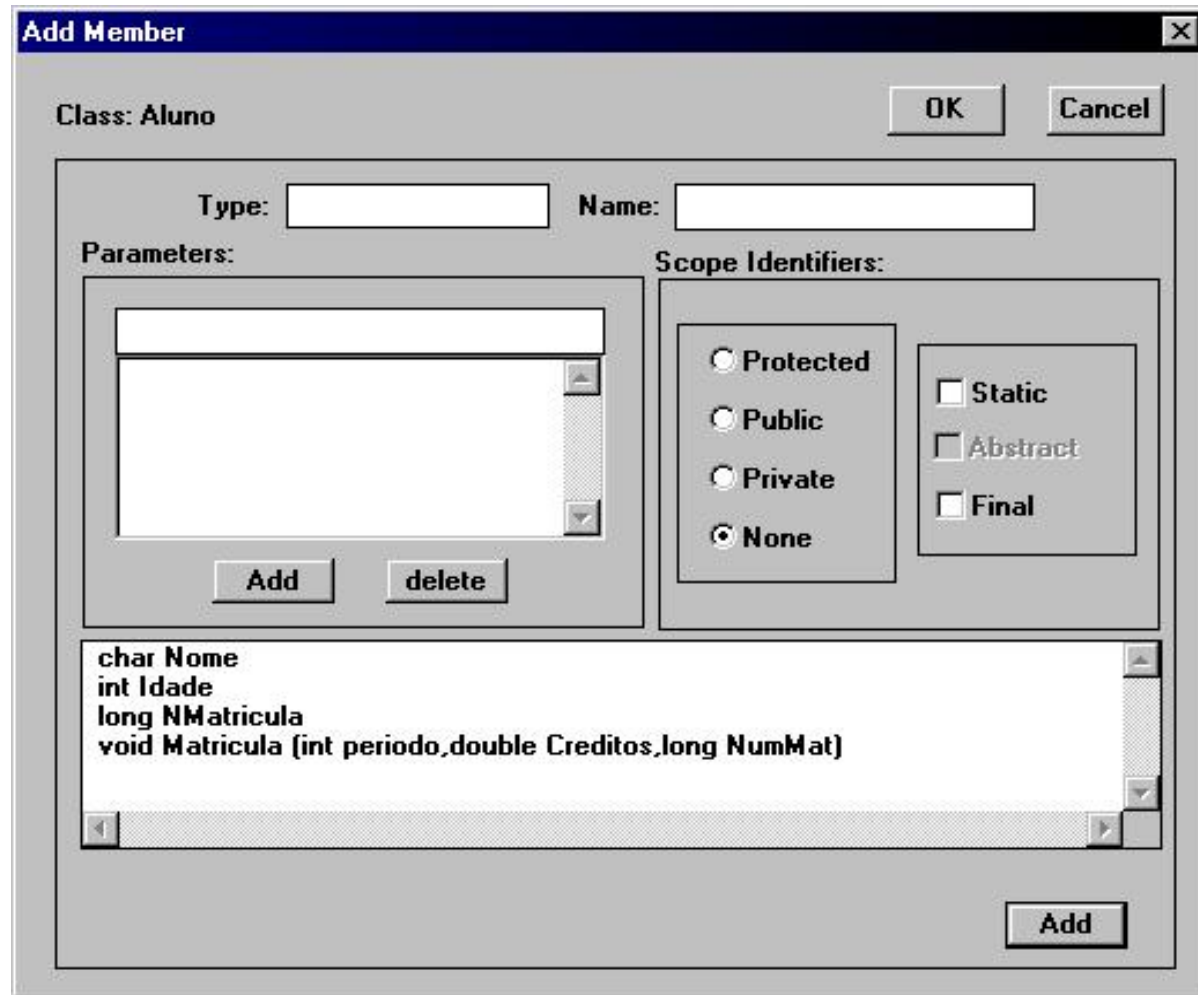
Implementation:  (.java)



# Movimentação de Classe



# Inserção de Membros



The image shows a Java Swing dialog box titled "Add Member" with a standard window control bar (minimize, maximize, close). The dialog is for adding a member to a class named "Aluno".

**Class:** Aluno

**Buttons:** OK, Cancel

**Type:** [Text Field] **Name:** [Text Field]

**Parameters:**

- [Text Area for parameters]
- Add** button
- delete** button

**Scope Identifiers:**

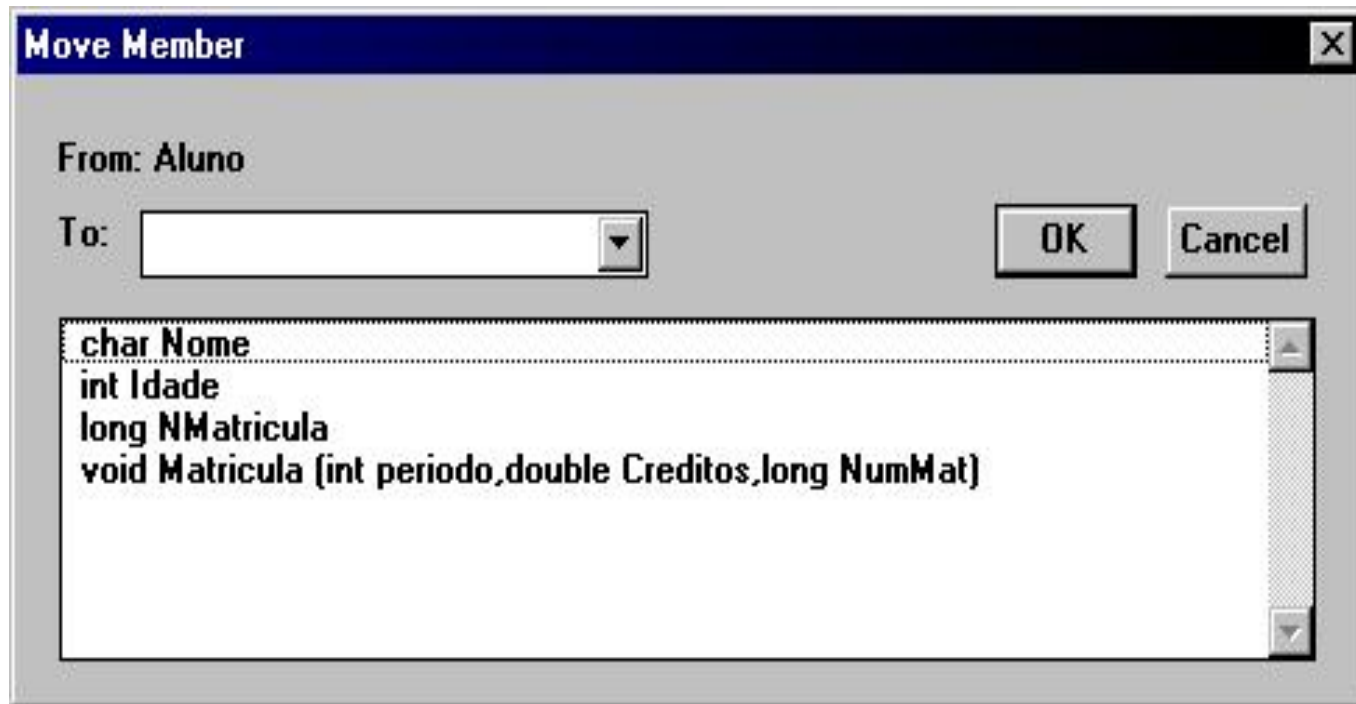
- ☐ Protected
- ☐ Public
- ☐ Private
- ☒ None
- ☐ Static
- ☐ Abstract
- ☐ Final

**Member List:**

```
char Nome  
int Idade  
long NMatricula  
void Matricula (int periodo,double Creditos,long NumMat)
```

**Add** button

# Movimentação de Membros



A screenshot of a 'Move Member' dialog box. The dialog has a title bar with the text 'Move Member' and a close button (X). Inside, there is a 'From:' label followed by the text 'Aluno'. Below this is a 'To:' label followed by a text input field with a dropdown arrow. To the right of the input field are two buttons: 'OK' and 'Cancel'. At the bottom of the dialog is a large text area containing the following text:

```
char Nome  
int Idade  
long NMatricula  
void Matricula (int periodo,double Creditos,long NumMat)
```

# Alteração da Classe e Membros

The image shows a 'Change Interface' dialog box for a class named 'Aluno' which inherits from 'Pessoa'. The dialog is divided into several sections. At the top, it shows the class name and base class. Below this, there are checkboxes for 'Abstract' and 'Final', and radio buttons for access modifiers: 'Public' (selected), 'Private', 'Protected', and 'None'. A central list box displays the class members: 'int Idade', 'char Nome', 'long NMatricula', and 'void Matricula (int periodo, double Creditos, long NumMat)'. Below the list box, there is a 'Type:' field with 'int' entered, and another set of checkboxes for 'Abstract', 'Static', and 'Final'. To the right of these is a 'Parameter:' section with an empty list box and 'Add' and 'Delete' buttons. At the bottom right are 'OK' and 'Cancel' buttons.

**Change Interface**

Class: **Aluno**  
Base: **Pessoa**

☐ Abstract  
☐ Final

☒ Public ☐ Private  
☐ Protected ☐ None

**int Idade**  
char Nome  
long NMatricula  
void Matricula (int periodo, double Creditos, long NumMat)

Type:

☐ Abstract ☐ Static  
☐ Final

☒ Public ☐ Private  
☐ Protected ☒ None

Parameter:

Add Delete

OK Cancel

# Generalização

**Generalize** [X]

Class: **Aluno** [OK] [Cancel]

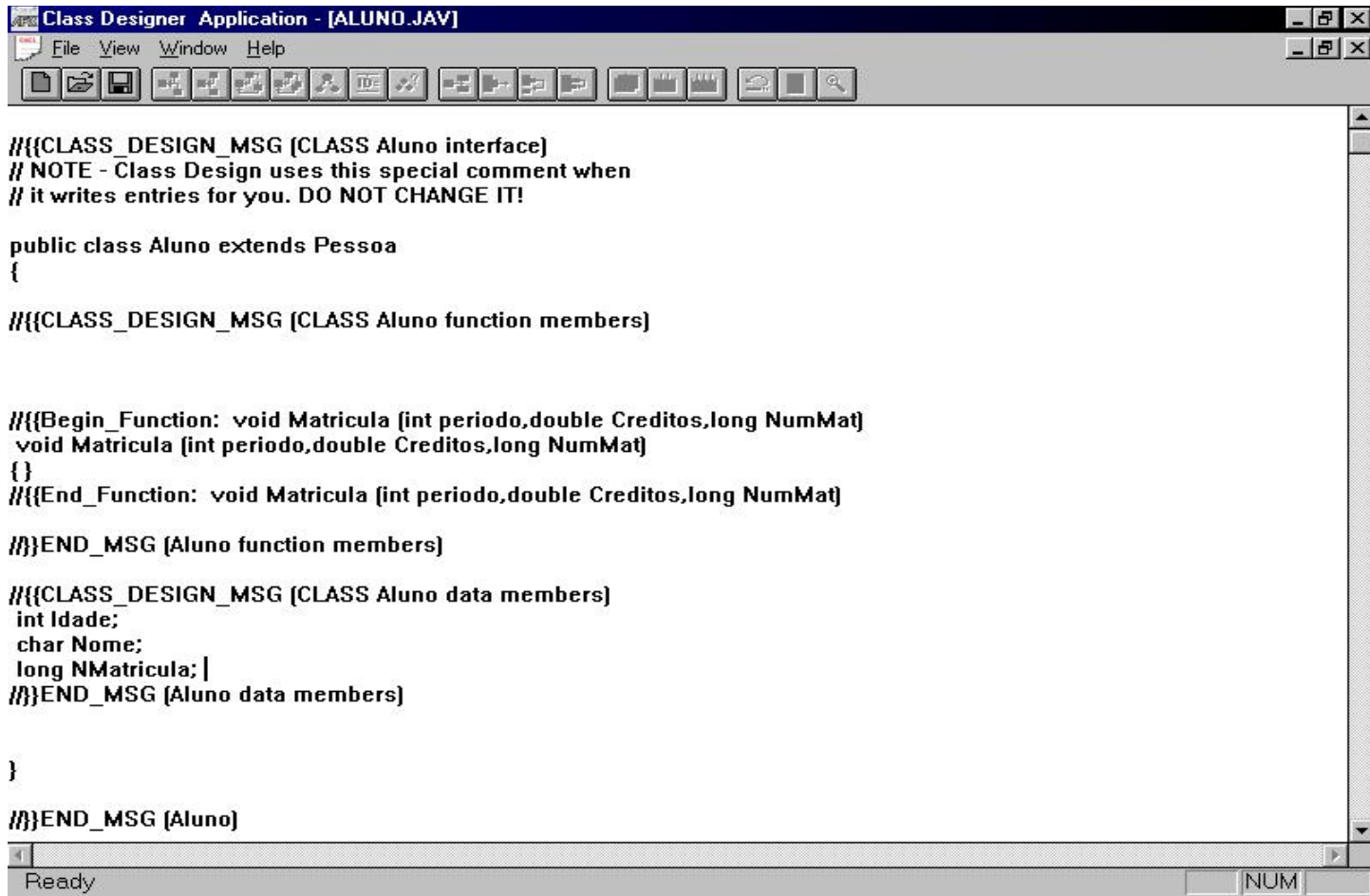
New class: [ ] [ ]

char Nome  
int Idade  
long NMatricula  
void Matricula (int periodo, double Creditos, long

☒ Automatic  
☒ Abstract function

[Add] [delete]

# Edição do Arquivo da Classe



The screenshot shows a window titled "Class Designer Application - [ALUNO.JAV]". The menu bar includes "File", "View", "Window", and "Help". Below the menu is a toolbar with various icons for file operations and design. The main text area contains the following Java code:

```
///CLASS DESIGN MSG (CLASS Aluno interface)  
// NOTE - Class Design uses this special comment when  
// it writes entries for you. DO NOT CHANGE IT!  
  
public class Aluno extends Pessoa  
{  
  
    ///CLASS DESIGN MSG (CLASS Aluno function members)  
  
    ///(Begin_Function: void Matricula (int periodo,double Creditos,long NumMat)  
    void Matricula (int periodo,double Creditos,long NumMat)  
    {}  
    ///(End_Function: void Matricula (int periodo,double Creditos,long NumMat)  
  
    ///END_MSG (Aluno function members)  
  
    ///CLASS DESIGN MSG (CLASS Aluno data members)  
    int Idade;  
    char Nome;  
    long NMatricula;  
    ///END_MSG (Aluno data members)  
  
}  
  
///END_MSG (Aluno)
```

At the bottom of the window, there is a status bar with the text "Ready" on the left and a button labeled "NUM" on the right.



# Conclusão

- Os objetivos do projeto foram concluídos com êxito
- Sugestões de aprimoramento
  - Possibilidade de troca de informações entre projetos distintos
  - Permitir a compilação de uma classe dentro do Class Designer Java
  - Possibilidade de se trabalhar com arquivos maiores



# Conclusão

- Sugestões de aprimoramento
  - Permitir a inclusão de interface para uma classe e exibi-la na janela principal
  - Trabalhar com membros na janela principal do Class Designer Java





**FIM**