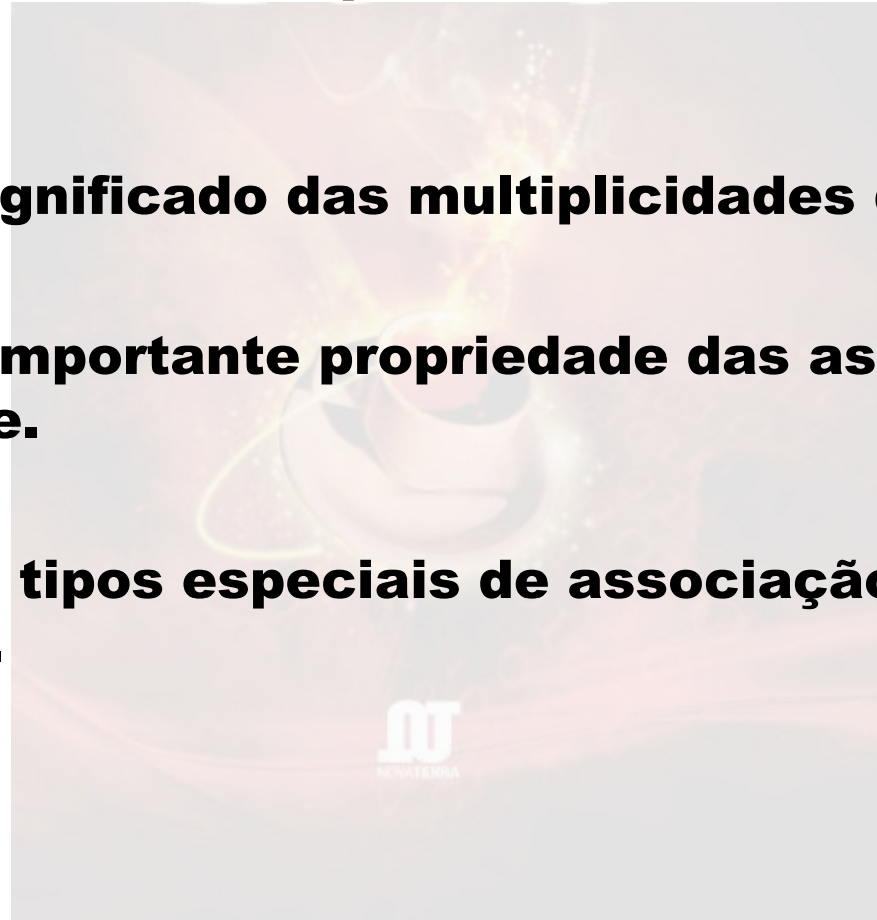


Capítulo 22

Associações entre Classes

Objetivos do Capítulo

- Indicar os diferentes aspectos de um relacionamento entre classes que podem ser expressos através de uma associação.**
- Descrever o significado das multiplicidades das associações.**
- Analisar uma importante propriedade das associações: a navegabilidade.**
- Distinguir dois tipos especiais de associação: a agregação e a composição.**



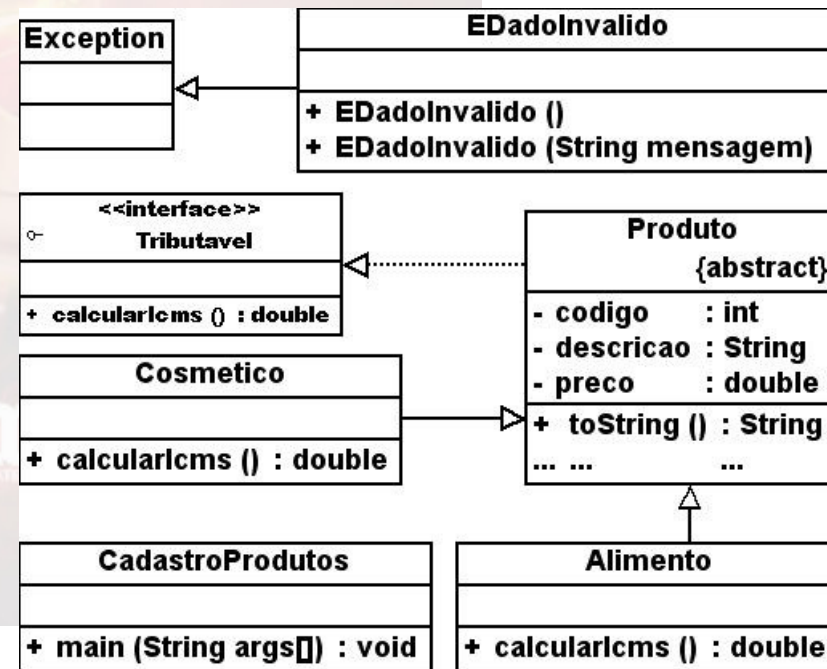
Introdução

❑ Sistema = conjunto de componentes

- Classes
- Interfaces

❑ Modelagem estrutural: diagrama de classes

- Componentes
- Relacionamentos
 - ❖ Generalização
 - ❖ Realização
 - ❖ Associação



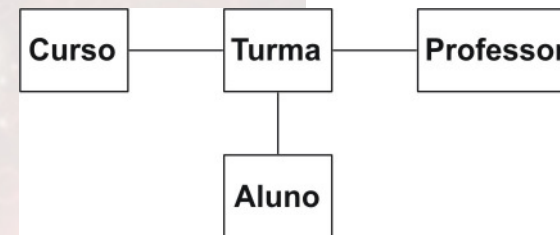
Representação Gráfica

❑ Linguagem: UML

- **Linha contínua**

❑ Exemplo: sistema para escolas de informática

- **Cadastrar cursos**
- **Abrir turmas**
- **Cadastrar professores**
- **Cadastrar alunos**
- **Registrar o professor da turma**
- **Matricular alunos na turma**

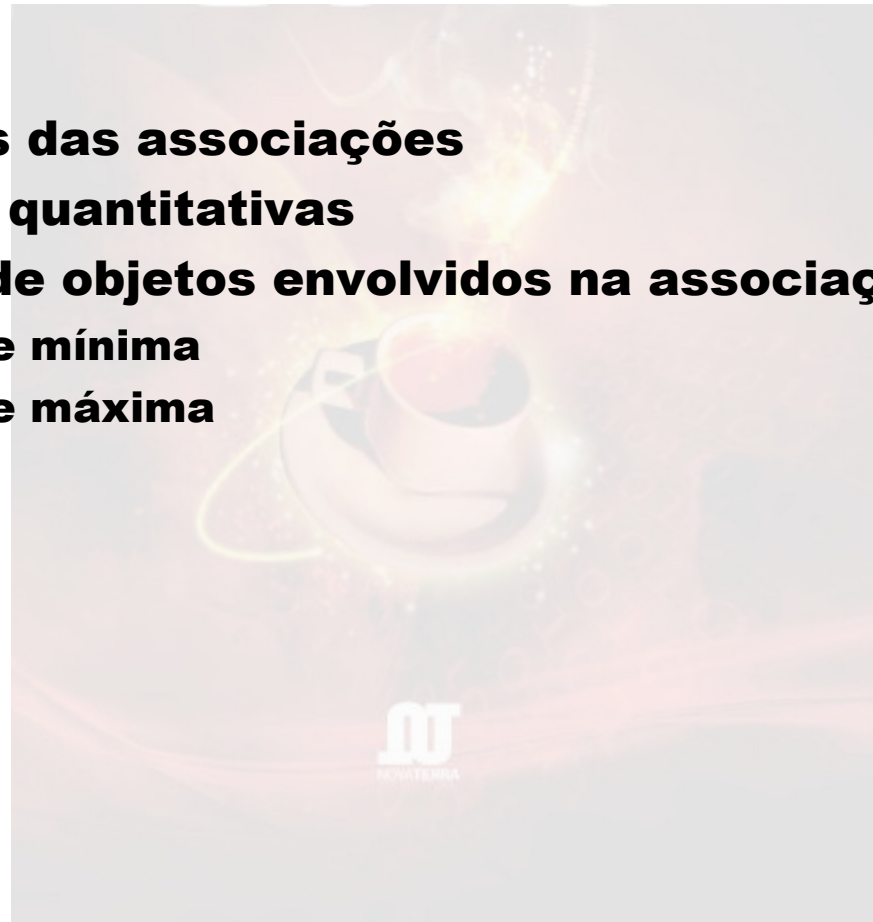


Multiplicidades

❑ Sinônimo: cardinalidades

❑ Conceito

- **Propriedades das associações**
- **Informações quantitativas**
- **Quantidade de objetos envolvidos na associação**
 - ❖ **Quantidade mínima**
 - ❖ **Quantidade máxima**



Multiplicidades

❑ Exemplo: comércio de livros

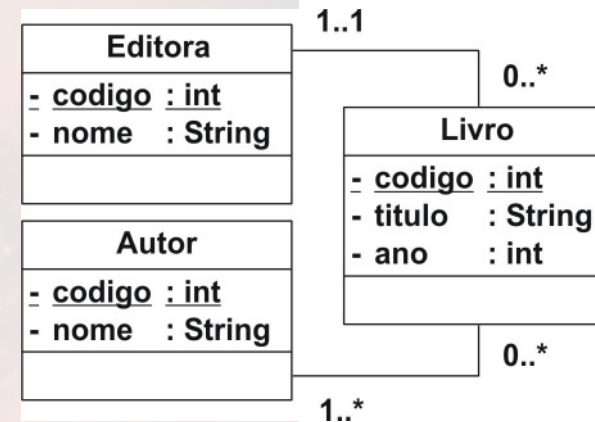
- Atributos identificadores
- Associação um-para-muitos
- Associação muitos-para-muitos

❑ Código correspondente

```
public class Editora {  
    private int codigo;  
    private String nome;  
}
```

```
public class Autor {  
    private int codigo;  
    private String nome;  
}
```

```
public class Livro {  
    private int codigo;  
    private String titulo;  
    private int ano;  
}
```



Navegabilidade

❑ Conceito

- **Propriedade da associação**
- **Representada por uma seta**
- **Indica que objetos de uma classe mantêm referências de objetos da classe associada**

❑ Tipos:

- **Unidirecional**
- **Bidirecional**

❑ Papel: a função que um objeto de uma classe exerce em relação aos objetos da classe associada

❑ Navegabilidade & Multiplicidade

- **Navegação para um objeto: uma única referência**
- **Navegação para múltiplos objetos: coleção de referências**

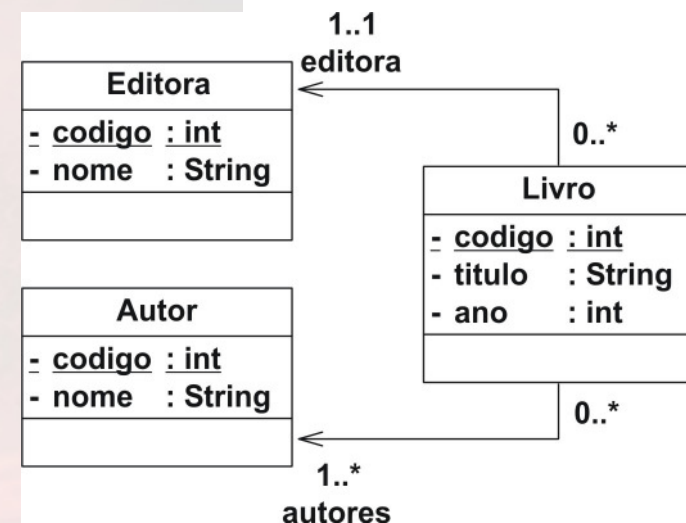
Navegabilidade

❑ Exemplo: comércio de livros

- Navegação de Livro para Editora: uma referência
- Navegação de Livro para Autor: múltiplas referências

❑ Código correspondente

```
1 import java.util.Collection;
2
3 public class Livro {
4     private int codigo;
5     private String titulo;
6     private int ano;
7     private Editora editora;
8     private Collection<Autor> autores;
9 }
```

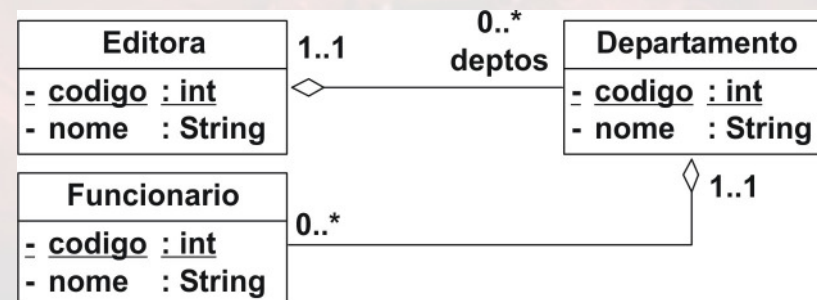


❑ Obs.: coleções mais específicas podem ser usadas.

Agregação

□ Conceito

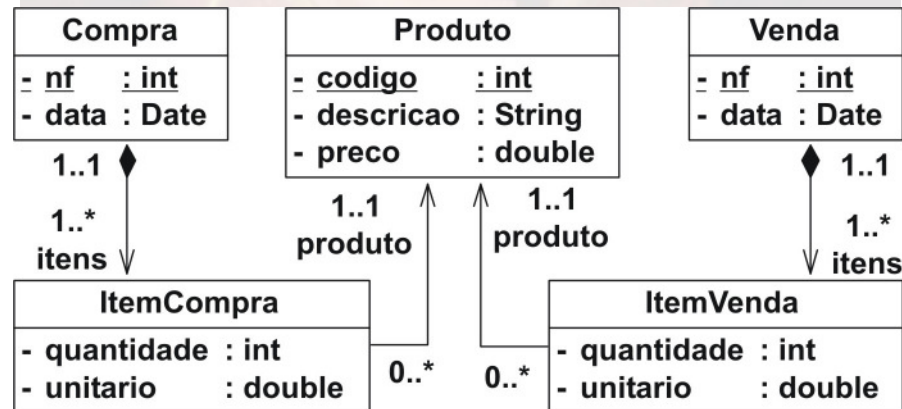
- **Tipo especial de associação**
- **Relação todo-parte: uma classe representa o todo e outra representa a parte deste todo**
- **Independência da parte: os objetos que representam a parte são utilizados em operações do sistema que não envolvem a classe que representa o todo**
- **Representação: losango vazio ao lado do todo**



Composição

❑ Conceito

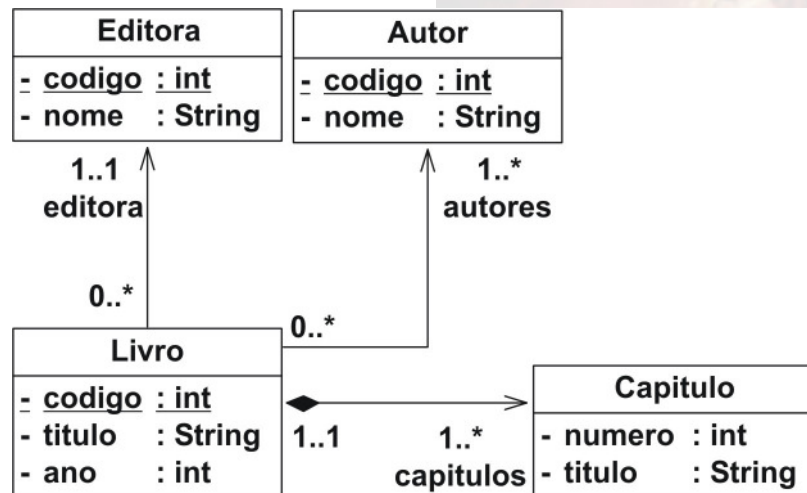
- **Tipo especial de associação**
- **Relação todo-parte: uma classe representa o contêiner e outra representa os elementos que ele abriga**
- **Dependência do conteúdo: os objetos que representam os elementos de um contêiner não fazem sentido fora dele**
 - ❖ **Se o contêiner é destruído os seus elementos também são**
- **Representação: losango preenchido do lado do contêiner**



Estudo de Caso

□ Ordem de implementação

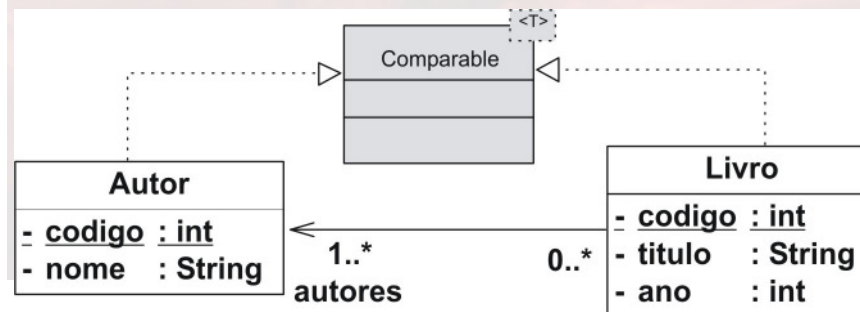
- **Editora.java**
- **Autor.java**
- **Capitulo.java**
- **Livro.java**
- **CadastroLivros.java**



```
C:\PCJ\Cap22>java CadastroLivros
1 - Java - 2010 - NovaTerra
Autores: [Rui Rossi dos Santos]
Capítulos: 4
    1) Introdução
    2) Ambiente
    3) Aplicativos
    4) Variáveis
2 - Mente - 1996 - Episteme
Autores: [Carlos Abreu][Fábio Santiago][Reginaldo Gomes]
Capítulos: 0
3 - Corpo - 2002 - Episteme
Autores: [Carla Pitanga][Mara Moreira]
Capítulos: 0
```

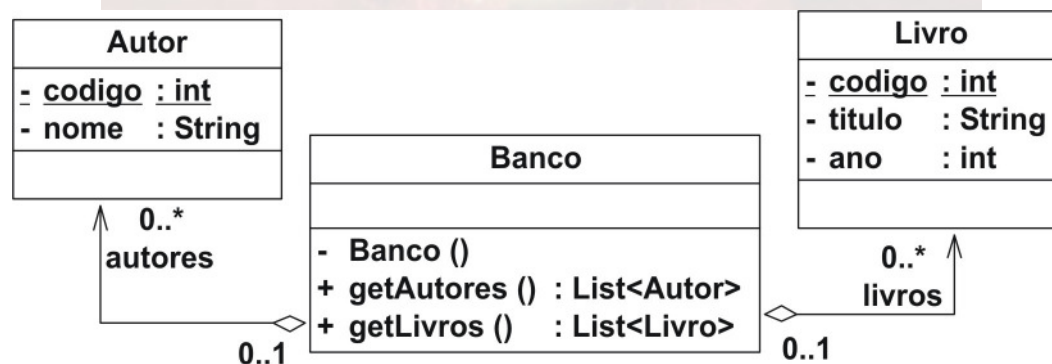
Exercício 1

- ❑ Copie as classes Autor e Livro para um diretório distinto e revise-as para que correspondam às especificações apresentadas na figura abaixo.
- ❑ Note que as classes Autor e Livro devem realizar a interface Comparable.
 - O critério de ordenação que deve ser utilizado para os autores é o nome e o critério de ordenação dos livros deve ser o título.
- ❑ Revise a classe Livro e elimine todas as referências que eram feitas às classes Editora e Capitulo.
 - Os atributos relacionados àquelas classes devem ser eliminados e os métodos devem ser eliminados ou adaptados.



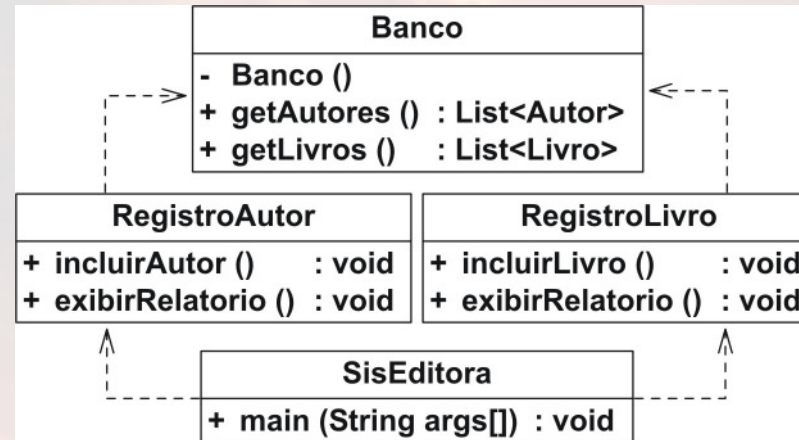
Exercício 2

- ❑ **Crie uma nova classe, chamada Banco, que agregue a ela uma lista de autores e uma lista de livros.**
 - **Implemente esta classe de acordo com a figura abaixo.**
- ❑ **Note que a classe Banco possui um único construtor e ele é privado. Desse modo, não é possível instanciar esta classe. A sua função será oferecer acesso às listas de autores e de livros e este acesso poderá ser realizado através dos métodos `getAutores()` e `getLivros()`. Antes de produzir o retorno, estes métodos devem verificar se as listas já foram criadas. Caso elas ainda estejam nulas, eles devem criá-las.**



Exercício 3

- ❑ Crie três novas classes, chamadas **RegistroAutor**, **RegistroLivro** e **SisEditora**.
 - Implemente estas classes de acordo com a figura abaixo.

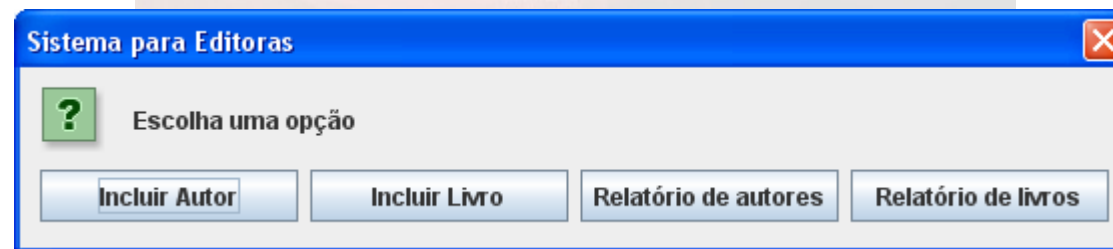


Exercício 3

- ❑ **A classe RegistroAutor deve permitir a manipulação dos cadastros de autores.**
 - **Seu método incluirAutor() deve solicitar o código e o nome de um autor e deve incluir o novo registro na lista de autores que se encontra na classe Banco.**
 - **O método exibirRelatorio() deve ordenar a lista de autores supracitada, recuperar os dados de todos os autores cadastrados e exibi-los através de um diálogo gráfico.**
- ❑ **A classe RegistroLivro deve permitir a manipulação dos cadastros de livros.**
 - **O método incluirLivro() deve solicitar o código, título, ano de publicação e autores do livro e deve gravar o novo registro na lista de livros que se encontra na classe Banco.**
 - **O método exibirRelatorio() deve ordenar a lista de livros, recuperar os dados de todos os livros cadastrados e apresentá-los através de um diálogo gráfico.**

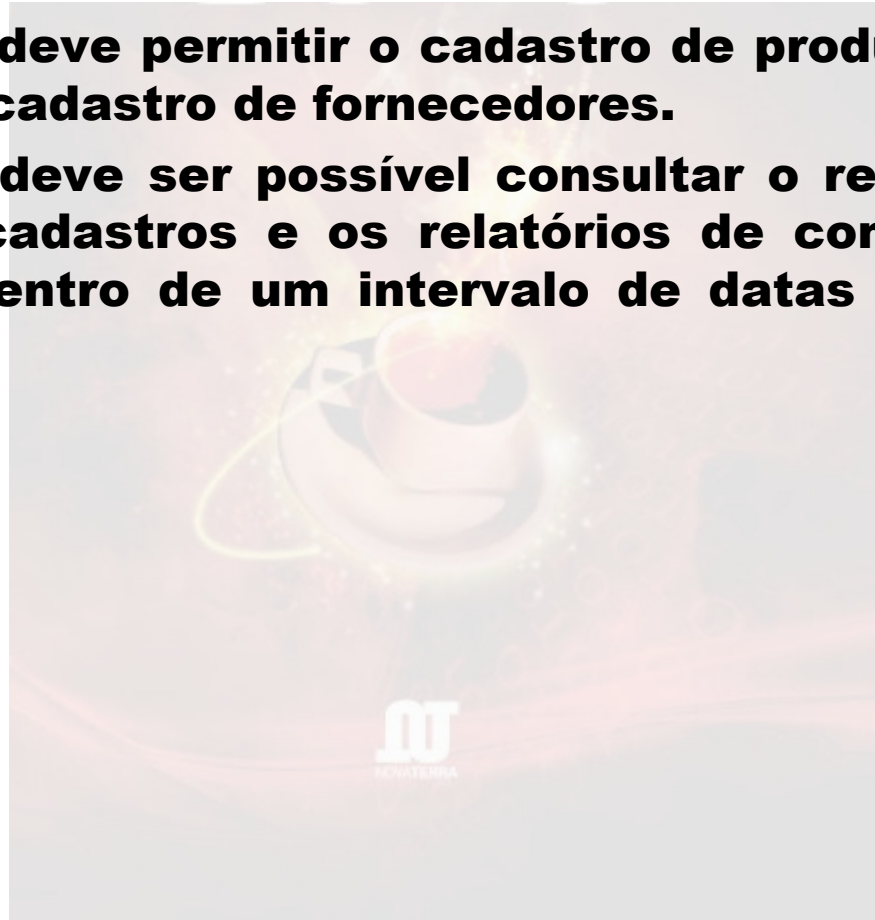
Exercício 3

- ❑ **A classe SisEditora deve produzir um diálogo contendo quatro opções. A figura abaixo ilustra a sua aparência.**
 - **Quando a opção “Incluir Autor” for selecionada, deve ser criado um objeto anônimo da classe RegistroAutor e seu método incluirAutor() deve ser invocado.**
 - **Quando a opção “Incluir Livro” for selecionada, deve ser criado um objeto anônimo da classe RegistroLivro e seu método incluirLivro() deve ser invocado.**
 - **Quando a opção “Relatório de autores” for selecionada, deve ser criado um objeto anônimo da classe RegistroAutor e seu método exibirRelatorio() deve ser invocado.**
 - **Quando a opção “Relatório de livros” for selecionada, deve ser criado um objeto anônimo da classe RegistroLivro e seu método exibirRelatorio() deve ser invocado.**



Exercício 4

- ❑ **Crie um novo aplicativo que permita registrar as compras e as vendas de produtos.**
 - **Ele também deve permitir o cadastro de produtos, o cadastro de clientes e o cadastro de fornecedores.**
 - **Além disso, deve ser possível consultar o relatório de qualquer um destes cadastros e os relatórios de compras e de vendas realizadas dentro de um intervalo de datas especificadas pelo usuário.**



Contato

Com o autor:

Rui Rossi dos Santos

E-mail: livros@ruirossi.pro.br

Web Site: <http://www.ruirossi.pro.br>

Com a editora:

Editora NovaTerra

Telefone: (21) 2218-5314

Web Site: <http://www.editoranovatterra.com.br>

