

INSTITUTO FEDERAL
Goiás

Instituto Federal de Goiás
Câmpus Goiânia

Bacharelado em Sistemas de Informação
Disciplina: Programação Orientada a Objetos I

Associação de Classes

Associações Bidirecionais

Prof. Ms. Dory Gonzaga Rodrigues
Goiânia - GO

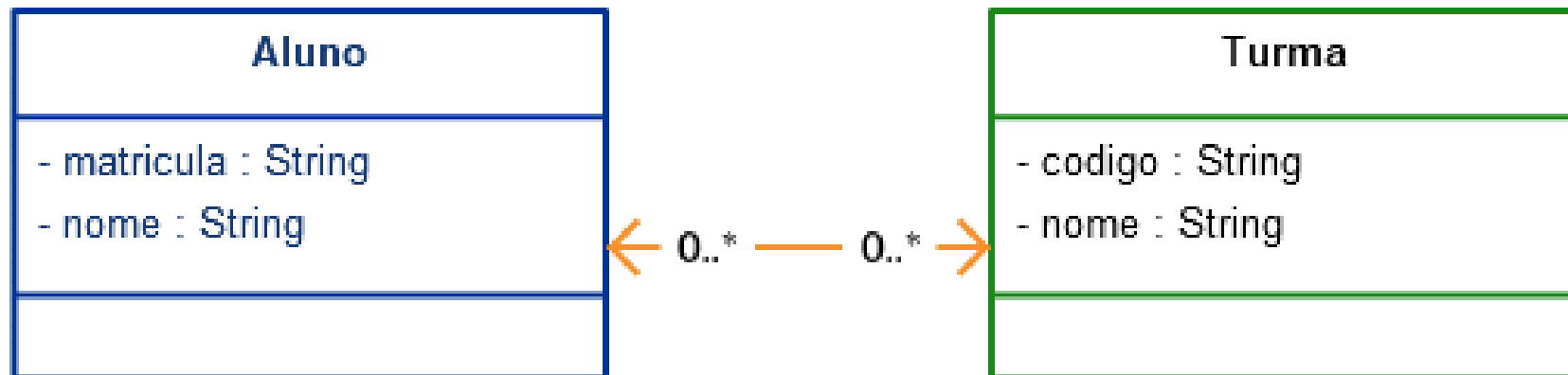
Associações Bidirecionais

As associações bidirecionais tem por objetivo permitir que o relacionamento seja conhecido por ambas as classes envolvidas



As associações bidirecionais aumentam o acoplamento (dependência entre classes), reduzindo a reusabilidade.

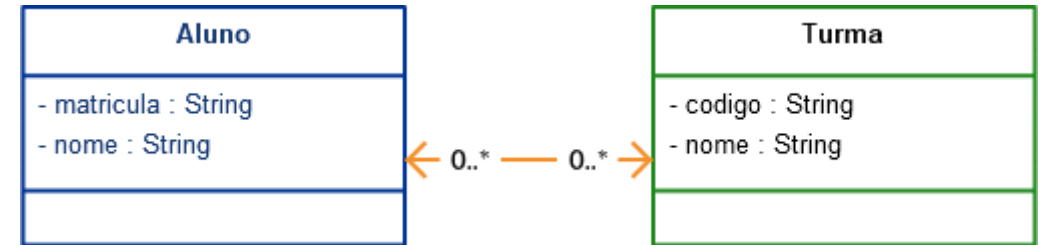
- ▶ Quando definidas como muitos para muitos, aumentam ainda mais a complexidade da Implementação.



Associações Bidirecionais

Implementando a Classe Aluno

```
1 import java.util.ArrayList;
2 import java.util.List;
3 import java.util.Objects;
4
5 public class Aluno {
6
7     private String matricula;
8     private String nome;
9     List<Turma> listaTurma = new ArrayList<>();
10
11     public Aluno(String nome) {
12         this.nome = nome;
13     }
14 }
```

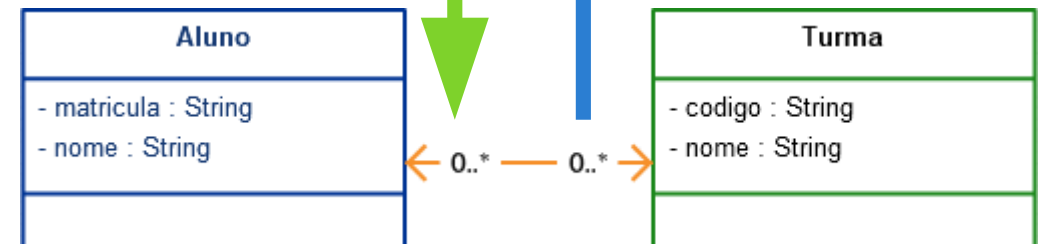


Um objeto aluno irá conter a relação das suas turmas (Lista)

Associações Bidirecionais

Implementando a Classe Aluno

```
15 public boolean adicionaTurma (Turma t) {  
16  
17     boolean sucesso = false;  
18  
19     if (t != null && !listaTurma.contains(t)) {  
20  
21         listaTurma.add(t);  
22         sucesso = true;  
23  
24         if (!t.listaAluno.contains(this)) {  
25             t.adicionaAluno(this);  
26         }  
27     }  
28     return sucesso;  
29 }
```



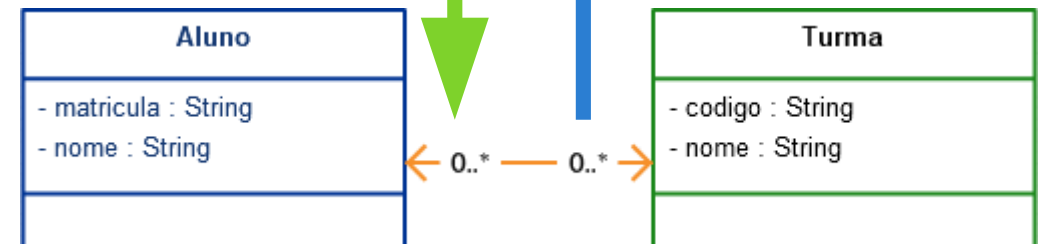
Ao adicionar uma turma em Aluno, deve ser adicionado também o aluno na lista em Turma

CONTINUA >>

Associações Bidirecionais

Implementando a Classe Aluno

```
31 public boolean removeTurma (Turma t) {  
32  
33     boolean sucesso = false;  
34  
35     if (!listaTurma.isEmpty() && listaTurma.contains(t)) {  
36         listaTurma.remove(t);  
37         sucesso = true;  
38  
39         if (t.listaAluno.contains(this)){  
40             t.removeAluno(this);  
41         }  
42     }  
43     return sucesso;  
44 }
```



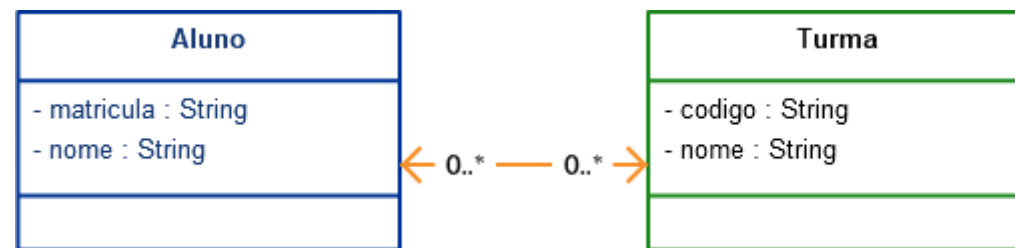
Ao remover uma turma de um Aluno, deve ser removido também o aluno da lista em Turma

CONTINUA >>

Associações Bidirecionais

Implementando a Classe Aluno

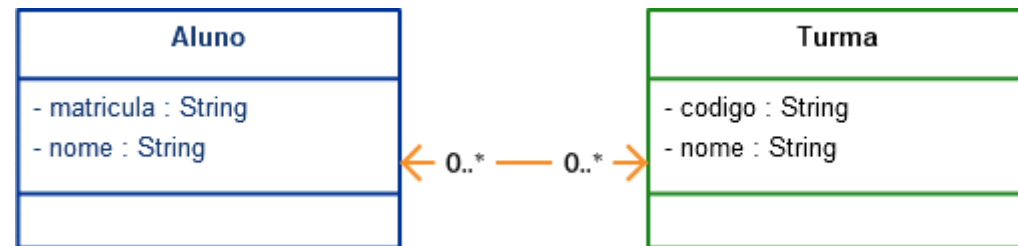
```
46 @Override
47 public boolean equals(Object obj) {
48     if (this == obj) {
49         return true;
50     }
51     if (obj == null) {
52         return false;
53     }
54     if (getClass() != obj.getClass()) {
55         return false;
56     }
57     final Aluno other = (Aluno) obj;
58     if (!Objects.equals(this.nome, other.nome)) {
59         return false;
60     }
61     return true;
62 }
```



Associações Bidirecionais

Implementando a Classe Aluno

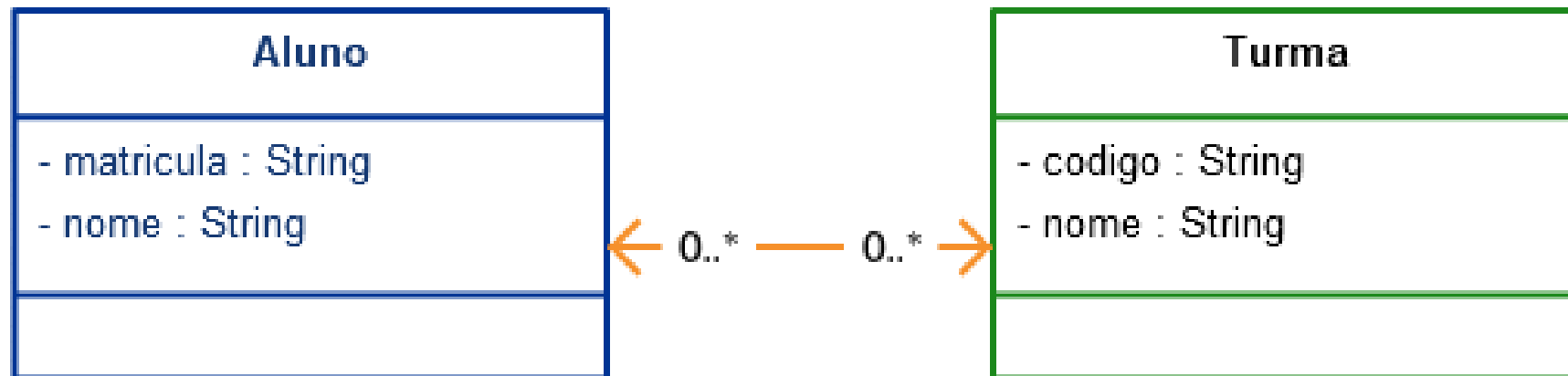
```
64 public String getNome() {  
65     return nome;  
66 }  
67  
68 public List<Turma> getListaTurma() {  
69     return listaTurma;  
70 }  
71  
72 @Override  
73 public String toString() {  
74     return "Aluno [nome: " + nome + " | matricula=" + matricula + "];"  
75 }  
76  
77 public void ImprimirAlunoTurma(){  
78  
79     System.out.println();  
80     System.out.println(this.toString());  
81  
82     for (Turma iTurma : this.getListaTurma()){  
83         System.out.println("-- " + iTurma.toString() );  
84     }  
85 }  
86 }
```



Associações Bidirecionais Implementando a Classe Turma



classe Turma e a relação com Aluno



Associações Bidirecionais

Implementando a Classe Teste

```
1 package aula5;
2
3 public class TesteTurmaAluno {
4     public static void main(String[] args){
5
6         Turma t1 = new Turma("POOI-2017");
7         Turma t2 = new Turma("BDII-2017");
8
9         Aluno a1 = new Aluno("Brenna");
10        Aluno a2 = new Aluno("Caio");
11        Aluno a3 = new Aluno("Raul");
12        Aluno a4 = new Aluno("Raissa");
13
14        t1.adicionaAluno(a1);
15        t1.adicionaAluno(a2);
16        t1.adicionaAluno(a3);
17        t1.adicionaAluno(a4);
18
19        t2.adicionaAluno(a3);
20        t2.adicionaAluno(a4);
21
22        t1.ImprimirTurmaAluno();
23        t2.ImprimirTurmaAluno();
24
25        a1.ImprimirAlunoTurma();
26        a3.ImprimirAlunoTurma();
27    }
28 }
```

A Saída do Programa deve ser assim:

```
Saída - Aula5 (run) X
run:
Turma [nome: POOI-2017 | codigo=null]
-- Aluno [nome: Brenna | matricula=null]
-- Aluno [nome: Caio | matricula=null]
-- Aluno [nome: Raul | matricula=null]
-- Aluno [nome: Raissa | matricula=null]

Turma [nome: BDII-2017 | codigo=null]
-- Aluno [nome: Raul | matricula=null]
-- Aluno [nome: Raissa | matricula=null]

Aluno [nome: Brenna | matricula=null]
-- Turma [nome: POOI-2017 | codigo=null]

Aluno [nome: Raul | matricula=null]
-- Turma [nome: POOI-2017 | codigo=null]
-- Turma [nome: BDII-2017 | codigo=null]
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```