

```
1 package classes.departamento;
2
3 import classes.curso.Curso;
4
5 public class Departamento {
6     private String nome;
7     private Curso curso; // Composição (Departamento-Curso)
8
9     public Departamento() {
10    }
11
12     public Departamento(String nomeDpto, Curso curso) {
13         this.nome = nomeDpto;
14         this.curso = curso;
15     }
16
17     public String getNome() {
18         return nome;
19     }
20
21     public void setNome(String nome) {
22         this.nome = nome;
23     }
24
25     public Curso getCurso() {
26         return curso;
27     }
28
29     @Override
30     public String toString() {
31         String dadosDpto = "Departamento: " + nome;
32
33         if (curso != null) {
34             dadosDpto += "\n" + curso.toString();
35         } else {
36             dadosDpto += "\n -> Nenhum curso associado (Composição)";
37         }
38         return dadosDpto;
39     }
40 }
```

```
1 import classes.aluno.Aluno;
2 import classes.curso.Curso;
3 import classes.departamento.Departamento;
4
5 public class Main {
6     public static void main(String[] args) {
7         System.out.println("---- Teste de Associações (Composição e Agregação) ----");
8
9         // 1. Criação e associação das classes
10        Aluno aluno1 = new Aluno(nome:"Joana Silva", matricula:"TADS2025");
11
12        // 2. Cria o Curso (lado independente da Agregação) e o associa ao Aluno
13        Curso cursoTADS = new Curso(nome:"Tecnologia em Análise e Desenv. de Sistemas", codigo:"C001");
14        cursoTADS.setAluno(aluno1);
15
16        // 3. Cria o Departamento (lado principal da Composição) e o associa ao Curso
17        Departamento dptoComputacao = new Departamento(nomeDpto:"Departamento de Ciência da Computação",
18        cursoTADS);
19
20        // 4. Impressão final de todos os dados através da classe principal (Departamento)
21        System.out.println("\n--- Impressão dos Dados ---");
22        System.out.println(dptoComputacao.toString());
23
24        // Teste de Agregação: Aluno existe independentemente do Curso.
25        System.out.println("\n--- Teste de Independência do Aluno (Agregação) ---");
26        System.out.println("Aluno original ainda existe: " + aluno1.getNome() + " / Matrícula: " + aluno1.getMatricula());
27    }
28 }
```

EXPLORER

src

- classes
  - aluno
    - Aluno.java

src > classes > aluno > Aluno

```
1 package classes.aluno;
2
3 public class Aluno {
4     private String nome;
5     private String matricula;
6
7     public Aluno() {
8     }
9
10    public Aluno(String nome, String matricula) {
11        this.nome = nome;
12        this.matricula = matricula;
13    }
14
15    public String getNome() {
16        return nome;
17    }
18
19    public void setNome(String nome) {
20        this.nome = nome;
21    }
22
23    public String getMatricula() {
24        return matricula;
25    }
26
27    public void setMatricula(String matricula) {
28        this.matricula = matricula;
29    }
30
31    @Override
32    public String toString() {
33        return " -> Aluno: " + nome + " (Matricula: " + matricula + ")";
34    }
35 }
```

EXPLORER

src

- classes
  - curso
    - Curso.java

src > classes > curso > Curso

```
5 public class Curso {
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22    public void setNome(String nome) {
23        this.nome = nome;
24    }
25
26    public String getCodigo() {
27        return codigo;
28    }
29
30    public void setCodigo(String codigo) {
31        this.codigo = codigo;
32    }
33
34    public Aluno getAluno() {
35        return aluno;
36    }
37
38    public void setAluno(Aluno aluno) {
39        this.aluno = aluno;
40    }
41
42    @Override
43    public String toString() {
44        String dadosCurso = "Curso: " + nome + " (Código: " + codigo + ")";
45
46        if (aluno != null) {
47            dadosCurso += "\n" + aluno.toString();
48        } else {
49            dadosCurso += "\n -> Nenhum aluno associado (Agregação)";
50        }
51        return dadosCurso;
52    }
53 }
```