

## Execução do Sistema — Classe Main

```
1  public class Main {
2      public static void main(String[] args) {
3
4          Disciplina d1 = new Disciplina("P358", "POO - Programação Orientada a Objetos", 1);
5          Disciplina d2 = new Disciplina("R255", "RLA - Raciocínio Lógico Algorítmico", 1);
6          Disciplina d3 = new Disciplina("S305", "SO - Sistemas Operacionais", 2);
7
8
9          Aluno a1 = new Regular("Luiz", "98765432101", 20, "A2124");
10         Aluno a2 = new Regular("Cíntia", "68100587125", 24, "A1422");
11         Aluno a3 = new Bolsista("Gabriel", "01234567890", 21, "A3243");
12
13         a1.apresentar();
14         a1.fezAniversario();
15
16         a2.apresentar();
17         a2.fezAniversario();
18
19         a3.apresentar();
20         a3.fezAniversario();
21
22         Professor p1 = new Professor("Andre", "9846510078453", 45, "Tecnologia");
23
24         p1.apresentar();
25         p1.fezAniversario();
26
27         Visitante v1 = new Visitante("Lucas", "8940354876", 32);
28         Visitante v2 = new Visitante("Pedro", "0002157489", 55);
29
30         v1.apresentar();
31         v2.apresentar();
32
33         Turma t1 = new Turma(d1, p1, "T1-001");
34         t1.adicionarAluno(a1);
35         t1.adicionarAluno(a2);
36         t1.adicionarAluno(a3);
37
38         Turma t2 = new Turma(d2, p1, "T2-002");
39         t2.adicionarAluno(a1);
40         t2.adicionarAluno(a2);
41         t2.adicionarAluno(a3);
42
43         Turma t3 = new Turma(d3, p1, "T3-003");
44         t3.adicionarAluno(a1);
45         t3.adicionarAluno(a2);
46         t3.adicionarAluno(a3);
47         t3.removerAluno(a2);
48
49         System.out.println("-----");
50         System.out.println(t1);
51         t1.listarAluno();
52
53         System.out.println("-----");
54         System.out.println(t2);
55         t2.listarAluno();
56
57         System.out.println("-----");
58         System.out.println(t3);
59         t3.listarAluno();
60
61     }
62 }
```

Na classe **Main** foi realizada a execução prática do projeto, instanciando objetos e demonstrando o funcionamento de todos os recursos implementados nas classes anteriores.

Inicialmente, foram criados três objetos da classe **Disciplina**, representando matérias pertencentes a diferentes períodos do curso. Em seguida, foram instanciados três objetos do tipo **Aluno**, sendo dois da subclasse **Regular** e um da subclasse **Bolsista**, aplicando assim o conceito de polimorfismo ao utilizar classes derivadas de Pessoa.

```
3
4     Disciplina d1 = new Disciplina("P358", "P00 - Programação Orientada a Objetos", 1);
5     Disciplina d2 = new Disciplina("R255", "RLA - Raciocínio Lógico Algorítmico", 1);
6     Disciplina d3 = new Disciplina("S305", "SO - Sistemas Operacionais", 2);
7
8
9     Aluno a1 = new Regular("Luiz", "98765432101", 20, "A2124");
10    Aluno a2 = new Regular("Cíntia", "68100587125", 24, "A1422");
11    Aluno a3 = new Bolsista("Gabriel", "01234567890", 21, "A3243");
12
```

```
12
13    a1.apresentar();
14    a1.fezAniversario();
15
16    a2.apresentar();
17    a2.fezAniversario();
18
19    a3.apresentar();
20    a3.fezAniversario();
21
```

Após a criação de cada aluno, os métodos **apresentar()** e **fazerAniversario()** foram chamados, exibindo suas informações iniciais e atualizando a idade conforme a lógica de encapsulamento definida em Pessoa.

Também foi instanciado um objeto da classe **Professor**, contendo suas credenciais e setor acadêmico, além de dois objetos da classe **Visitante**, que representam pessoas externas vinculadas de forma temporária à instituição.

```
21
22    Professor p1 = new Professor("Andre", "9846510078453", 45, "Tecnologia");
23
24    p1.apresentar();
25    p1.fezAniversario();
26
27    Visitante v1 = new Visitante("Lucas", "8940354876", 32);
28    Visitante v2 = new Visitante("Pedro", "0002157489", 55);
29
30    v1.apresentar();
31    v2.apresentar();
32
```

Posteriormente, foram criadas três **Turmas**, cada uma associada a uma disciplina e ao mesmo professor. Aos objetos Turma foram adicionados os alunos previamente criados, utilizando o método **adicionarAluno()**. Além disso, foi demonstrada a funcionalidade de **remoção** ao retirar um dos alunos da terceira turma.

```
32
33     Turma t1 = new Turma(d1, p1, "T1-001");
34     t1.adicionarAluno(a1);
35     t1.adicionarAluno(a2);
36     t1.adicionarAluno(a3);
37
38     Turma t2 = new Turma(d2, p1, "T2-002");
39     t2.adicionarAluno(a1);
40     t2.adicionarAluno(a2);
41     t2.adicionarAluno(a3);
42
43     Turma t3 = new Turma(d3, p1, "T3-003");
44     t3.adicionarAluno(a1);
45     t3.adicionarAluno(a2);
46     t3.adicionarAluno(a3);
47     t3.removerAluno(a2);
48
```

Por fim, o programa imprime no console as informações de cada turma, utilizando o método **toString()** sobrescrito, além da listagem detalhada dos alunos matriculados por meio do método **listarAluno()**, apresentando de forma clara e organizada os dados cadastrados.

```
48
49     System.out.println("-----");
50     System.out.println(t1);
51     t1.listarAluno();
52
53     System.out.println("-----");
54     System.out.println(t2);
55     t2.listarAluno();
56
57     System.out.println("-----");
58     System.out.println(t3);
59     t3.listarAluno();
60
61 }
62 }
```

Essa classe evidencia a funcionalidade geral do sistema, integrando todas as classes desenvolvidas e demonstrando o uso prático dos conceitos de **orientação a objetos**, como herança, encapsulamento, polimorfismo, composição e associação.