

```

1 package br.univali.kob.poo1.aula02a;
2
3 import java.time.LocalDate;
4
5 /**
6  * Estudante no contexto de um sistema acadêmico.
7  * <p>
8  * Em Java, utilizamos a palavra reservada "extends" para indicar que a
9  * classe atual (subclasse) está estendendo outra classe (superclasse).
10 * Aqui, a classe Student é uma subclasse de Person e, portanto, Person
11 * é a superclasse de Student. Em Java, uma classe pode estender somente
12 * uma única classe (permite apenas herança simples).
13 *
14 * @author Marcello Thiry
15 */
16 public class Student extends Person {
17
18     /**
19      * Número de matrícula do estudante.
20      */
21     private int rollNumber;
22
23     /**
24      * Data na qual o estudante foi admitido (primeira matrícula).
25      */
26     private LocalDate enrollmentDate;
27
28     /**
29      * Data na qual o estudante se desligou da instituição.
30      */
31     private LocalDate dropDate;
32
33     /**
34      * Simula um getter que informa se o aluno está ou não matriculado
35      * a partir da data de desligamento. Se o aluno ainda não tem uma data
36      * de desligamento, então é porque ele ainda está matriculado.
37      * <p>
38      * O método abaixo mostra uma boa prática de codificação para operações de
39      * acesso (acessor) que retornam valores booleanos. Tipicamente, utiliza-se
40      * o prefixo "is" (é ou está) ao invés do prefixo "get". No nosso caso, é um
41      * falso getter, pois ele utiliza outro atributo para verificar se o aluno
42      * está matriculado.
43      *
44      * @return a situação atual do estudante: matriculado (true) ou não (false).
45      */
46     public boolean isEnrolled() {
47         return dropDate == null;
48     }
49
50     /**
51      * Getter.
52      *
53      * @return a data na qual o estudante foi admitido (primeira matrícula)
54      */
55     public LocalDate getEnrollmentDate() {
56         return enrollmentDate;
57     }
58
59     /**
60      * Setter.
61      *
62      * @param enrollmentDate a data na qual o estudante foi admitido (primeira matrícula)
63      */
64     public void setEnrollmentDate(LocalDate enrollmentDate) {
65         this.enrollmentDate = enrollmentDate;
66     }
67
68     /**
69      * Getter.
70      *
71      * @return a data na qual o estudante se desligou da instituição, null se ainda matriculado
72      */
73     public LocalDate getDropDate() {

```

```
72         return dropDate;
73     }
74
75     /**
76      * Setter.
77      *
78      * @param dropDate a data na qual o estudante se desligou da instituição
79      */
80     public void setDropDate(LocalDate dropDate) {
81         this.dropDate = dropDate;
82     }
83
84     /**
85      * Getter.
86      *
87      * @return o número de matrícula do estudante
88      */
89     public int getRollNumber() {
90         return rollNumber;
91     }
92
93     /**
94      * Setter.
95      *
96      * @param rollNumber o número de matrícula do estudante
97      */
98     public void setRollNumber(int rollNumber) {
99         this.rollNumber = rollNumber;
100     }
101
102 }
```