

```

/**
 * A classe de Estudante representa um aluno em um sistema de
 * administração de alunos.
 * Ela mantém os detalhes de cada aluno relevantes em nosso
 * contexto.
 *
 * Versão em português do livro Programação orientada a objetos
 * com Java de Michael Kolling e David Barns
 *
 * @author Simone Santos
 * @version 26/01/2021
 */
public class Estudante {
    //o nome completo do aluno
    private String nome;
    //a ID do aluno
    private String id;
    //a quantidade de créditos para estudo obtida até agora
    private int credito;

    /**
     * Cria um novo aluno com nome e ID
     * @param nomeCompleto Nome do aluno
     * @param id ID do aluno
     */
    public Estudante(String nomeCompleto, String id) {
        this.nome = nomeCompleto;
        this.id = id;
        this.credito = 0;
    }

    /**
     * Retorna o nome completo do aluno
     * @return o nome
     */
    public String getNome() {
        return nome;
    }

    /**
     * Configura um novo nome para esse aluno
     * @return o nome
     */
    public void setNome(String novoNome) {
        nome = novoNome;
    }

    /**

```

```

    * Retorna o ID do estudante
    * @return o ID
    */
    public String getId() {
        return id;
    }

    /**
     * Adiciona alguns pontos de crédito aos
     * créditos acumulados pelo aluno.
     */
    public void addCredito(int novoCredito) {
        credito += novoCredito;
    }

    /**
     * Retorna o número de pontos de crédito que o
     * aluno acumulou.
     * @return o credito
     */
    public int getCreditos() {
        return credito;
    }

    /**
     * Retorna o nome de login desse aluno.
     * O nome de login é uma combinação dos primeiros quatro
     * caracteres do nome do aluno e dos primeiros três
     * caracteres do número de ID do aluno.
     */
    public String getLoginNome() {
        return nome.substring(0, 4) + id.substring(0, 3);
    }

    /**
     * Imprime o nome e o ID do aluno
     * no terminal
     */
    public void print() {
        System.out.println(nome + " (" + id + " )");
    }
}

```

Lista 6

Analise, implemente, crie objetos e chame métodos da classe `Estudante` e baseado nesta classe faça os exercícios a seguir.

1. O que seria retornado por `getLoginNome` para um aluno com nome “Henrique Moreira” e id “557214”?
2. Crie um objeto da classe `Estudante` com nome “dbj” e id “859012”. O que acontece quando `getLoginNome` é chamado sobre este aluno? Por que você acha isso?
3. A classe `String` define um método de acesso `length` com a seguinte assinatura:

```
/**
 * Retorna o número de caracteres nessa string.
 */
public int length()
```

Adicione instruções condicionais ao construtor de `Estudante` para imprimir uma mensagem de erro se o comprimento do parâmetro `nomeCompleto` tiver menos de quatro caracteres ou o comprimento do parâmetro `id` tiver menos de três caracteres. Mas o construtor ainda deve utilizar esses parâmetros para configurar os atributos `nome` e `id`, mesmo que a mensagem de erro seja impressa.

4. **Exercício de desafio.** Modifique o método `getLoginNome` de modo que gere sempre um nome de login, mesmo que qualquer um dos campos `nome` e `id` não sejam suficientemente longos. Para strings mais curtas do que o comprimento requerido, utilize a string inteira.

O **exercício de desafio** pode ser enviado como atividade extra que valerá pontos de participação. Envie um e-mail para simoneos.trab@gmail.com com o assunto “Exercício de desafio”.

Pode ser enviado apenas o método, seja no corpo do e-mail ou em um arquivo em anexo. Lembrando, quero apenas o método, não precisa enviar a classe inteira.