

Lista de exercícios 4 LPOO

1. Implemente e entenda os exemplos da aula;
2. **(Tarefa 5)** Escreva a estrutura de uma classe (atributos e métodos) para representar um aluno (nome, matrícula, curso, período, disciplinasMatriculadas(array de String) e endereço):
 - a. Compile a classe;
 - b. Crie um construtor para essa classe com a seguinte assinatura:
public Aluno(String nome, String matricula, String curso, int periodo, int idade, int quantidadeDisciplinasPermitidadas)
Onde a quantidadeDisciplinasPermitidadas representa a quantidade de disciplinas que o aluno pode se matricular. Deve ser utilizada para redimensionar o array de Strings.
 - c. Acrescente três métodos:
 - String fazMatricula(String disciplina): inclui uma disciplina no array de Strings. Se o aluno já ultrapassou a quantidade de disciplinas que pode se matricular, então o sistema retorna para este método uma string informando: "Quantidade de disciplinas excedida. O limite de disciplinas para este aluno é de 2 disciplina(s). Se desejar, cancele a matrícula de uma das disciplinas e faça a nova matrícula."
Se o aluno não puder ser matriculado em nenhuma disciplina (tamanho do array=0), então o método deve retornar: "Este aluno não pode ser matriculado em nenhuma disciplina, por favor, fale com a secretaria."
Se foi matriculado com sucesso a seguinte String é retornada: "Matrícula na disciplina "LPOO" executada."
 - String cancelarMatricula(String disciplina): exclui a disciplina do array de Strings. Se o aluno não estiver matriculado na disciplina, o método deve retornar: "Aluno não está matriculado na disciplina LPOO, portanto não é possível cancelar esta matrícula.". Se o aluno realmente estiver matriculado na disciplina, então o método deve retornar "Cancelamento da matrícula da disciplina LPOO executado com sucesso."
 - String imprime(): Método que retorna uma String de forma legível com todos os atributos de aluno e as disciplinas matriculadas no seguinte formato:

```
-----  
Nome do Aluno: Rafael Romualdo Wandresen  
Matricula: GRR20130101  
Curso: TADS  
Periodo: 4  
Disciplinas Matriculadas: Gestão de Projetos; LPOO  
-----
```

- d. Crie uma outra classe (SistemaAcademico) que utilize objetos do tipo Aluno. Nessa classe instancie um array de Alunos que armazene os alunos criados. Crie um menu com as seguintes opções: 1 – Cadastrar Aluno, 2 – Excluir aluno por nome, 3 – Listar Alunos, 4 – Matricular Aluno em Disciplina, 5 – Cancelar Matrícula, 6 – Imprimir lista Alunos e

Disciplinas Matriculadas. Cada um destes itens do menu deve ter um método associado com os seguintes métodos. Criar todos os métodos estáticos:

1. `public static void cadastrarAluno(Aluno aluno)`
 2. `public static void excluirAlunoPorNome(String nomeAluno)`
 3. `public static Aluno[] listarAlunos()`
 4. `public static String matricularAlunoEmDisciplina(Aluno aluno, String disciplina)`. Onde o retorno informa o que ocorreu com a matrícula do aluno, conforme especificado no método `fazMatricula`.
 5. `public static String cancelarMatricula(Aluno aluno, String disciplina)`
 6. `public static String imprimirListaDeAlunoseDisciplinas ()`
- e. Quando o sistema iniciar deve pedir ao usuário para informar a quantidade de alunos que serão cadastrados. Com essa informação dimensione o array.
- f. Quando o usuário optar pela opção 4, se for a primeira disciplina do aluno, o sistema deve perguntar em quantas disciplinas o aluno deve ser matriculado. Com essa informação dimensione o array de Strings.
3. **(Tarefa 5)** Altere o código da classe Ponto 3D para que os métodos tenham o modificador `public`, `protected` e `private`.
- a. Para cada uma das alterações tente compilar e rodar.
 - b. Em algum dos casos ocorreu erro de compilação ou execução? Quais e porque?
4. **(Tarefa 5)** Modifique a classe Bicicleta para encapsular seus atributos como privados, crie os métodos de acesso a estes atributos e crie dois construtores para ela:
- a. Um construtor que recebe como parâmetro o número máximo de marchas da bicicleta (crie um atributo privado para armazenar este valor)
 - b. Um construtor padrão (sem argumentos) que inicie o número máximo de marchas como 18.
 - c. Ambos os construtores devem criar uma bicicleta parada.
 - d. Os métodos que mudam a marcha devem avaliar se o número máximo de marchas foi alcançado e não permitir a mudança nestes casos.
5. Crie um programa que leia dois pares de valores reais (coordenadas) e, utilizando o objeto `Ponto2D`, mostre a distância entre as coordenadas;
6. Implemente a classe `Triângulo` representada abaixo. Crie um programa que leia os lados do triângulo e diga se ele é equilátero, isósceles ou escaleno, utilizando os métodos do objeto;

Triângulo
<code>int lado1</code> <code>int lado2</code> <code>int lado3</code>
<code><getters...></code> <code><setters...></code> <code>Triângulo (int l1, int l2, int l3)</code> <code>boolean ehEscaleno()</code> <code>boolean ehEquilatero()</code> <code>boolean ehIsosceles()</code>

7. Crie uma classe `Quadrado`, que seja capaz de armazenar o tamanho do lado do quadrado e informar seu perímetro e sua área. Crie um programa que demonstre a funcionalidade do objeto;

8. **(Tarefa 5)** Implemente uma classe contendo apenas métodos estáticos capazes de realizar todas as conversões de temperatura possíveis entre as unidades Celsius, Farenheit e Kelvin. São conhecidas as seguintes relações de conversão: $F = 9 \cdot C / 5 + 32$ e $K = C + 273$. Demonstre a utilização do objeto. Utilize, neste programa, uma estrutura de controle switch para identificar a conversão solicitada pelo usuário;
9. Faça o download da apostila da Caelum em:
<http://www.caelum.com.br/download/caelum-java-objetos-fj11.pdf>
 - a. Leia o capítulo 4
 - b. Faça os exercícios da página 44 até a página 47 (4.12 - Exercícios: Orientação a Objetos)