

<b>UNIFEI</b>	Universidade Federal de Itajubá Instituto de Engenharia de Sistemas e Tecnologias da Informação - IESTI
<b>5º Laboratório – Exercício 2</b>	<b>Disciplina Projeto de Software – Prof. Enzo Seraphim</b>

1) [Opcional se já feito] Instale o Dia Diagram Editor de licença GPLv2 através, fazendo o download em: <http://dia-installer.de>

2) Faça a leitura da especificação abaixo identificando as possíveis classes que representam o padrão de projeto Compositor

Elementos elétricos pode ser um circuito ou componentes. No circuito elétrico a corrente elétrica pode: percorrer um único caminho, chamado de série; percorrer mais de um caminho, chamado de paralelo; ter partes em série e partes em paralelo, chamado de misto. Quando no caminho a ser percorrido pela corrente elétrica existe uma interrupção o circuito tem o estado aberto e quando não existe impedimento algum o circuito está no estado fechado. O circuito elétrico é formado por componentes elétricos ou outro circuito elétrico. Os componentes elétricos podem ser resistor, capacitor, bobina e memristor. Um rótulo de identificação usado para diferenciar circuitos e para diferenciar componentes.

3) Modele o diagrama de classes de UML no Dia usando o padrão de projeto Compositor, sendo: ElementoEletrico como sendo a classe Componente do padrão; ComponenteEletrico como sendo a classe Folha do padrão; CircuitoEletrico como sendo a classe Compositor do padrão; App com main como sendo a classe Cliente do padrão. Adicione os atributos e os relacionamentos de acordo com a especificação. Omita métodos *gets* e *sets* para atributos privados da classe e os métodos manipuladores de relacionamentos entre as classes.

4) Crie uma classe chamada CaminhoEnum com esteriótipo enum para restringir os valores para o atributo caminho da classe CircuitoEletrico com os valores: UNICO e VARIOS. Crie outra classe chamada EstadoEnum com esteriótipo enum para restringir os valores para o atributo estado da classe CircuitoEletrico com os valores: ABERTO e FECHADO. Crie mais uma classe chamada ComponenteEnum com esteriótipo enum para restringir os valores para o atributo tipo da classe ComponenteEletrico com os valores: RESISTOR, CAPACITOR, BOBINA e MEMRISTOR. Finalmente, coloque esses enum como tipos dos atributos das classes correspondentes.

5) Para manipular parte ou toda hierarquia defina o método abstrato totalElementos() na classe abstrata ElementoEletrico. Esse método retorna um inteiro que representa a quantidade de elementos. Adicione o método totalElementos() mostrando sua implementação nas subclasses ComponenteEletrico e CircuitoEletrico.

6) Crie o pacote br.edu.unifei.ecot12.eletrico

7) Salve o diagrama UML no Dia como eletrico C:\Usuários\Aluno\vscode\ecot12-lab05-inicial

8) **Crie o pacote eletrico:**

- Botão direito br.edu.unifei.ecot12 e selecione a opção New Java Package com o nome: eletrico

11) **Crie classe abstract ElementoEletrico:**

- Clique com o botão esquerdo do mouse no pacote br.edu.unifei.ecot12.eletrica em Explorador do VSCode e selecione New Java File | Class
- Input the class name: ElementoEletrico
- Conforme diagrama UML, declare seu atributo rotulo.
- Declare atributos e crie os gets e sets.
- Declare o método abstrato totalElementos

12) **Crie subclasse ComponenteEletrico:**

- Clique com o botão esquerdo do mouse no pacote br.edu.unifei.ecot12.eletrica em Explorador do VSCode e selecione New Java File | Class
- Input the class name: ComponenteEletrico
- Declare atributos e crie os gets e sets.
- Implemente o método totalElementos para que retorne o valor 1.

13) **Crie subclasse CircuitoEletrico:**

- Clique com o botão esquerdo do mouse no pacote br.edu.unifei.ecot12.eletrica em Explorador do VSCode e selecione New Java File | Class
- Input the class name: ElementoEletrico

- Declare atributos e crie os gets e sets.

#### 14) **Implemente o método totalElementos da classe CircuitoEletrico:**

@Override

```
public int totalElementos() {
    int soma=1;
    for (ElementoEletrico e : elementos) {
        soma+=e.totalElementos();
    }
    return soma;
}
```

#### 15) **Crie uma classe App com main (Cliente):**

- Clique com o botão esquerdo do mouse no pacote br.edu.unifei.ecot12.eletrica em Explorador do VSCode e selecione New Java File | Class
- Input the class name: Main
- Declare o main : public static void main (String args[] ) {}

#### 16) **Implemente o método main de App:**

No main instancie os 8 objetos da classe Componente e os 4 objetos da classe Circuito. Conforme figura abaixo, adicione cada filho a seu respectivo pai para que obtenha a seguinte estrutura hierárquica.

