

UNIFEI	Universidade Federal de Itajubá
	Instituto de Engenharia de Sistemas e Tecnologias da Informação - IESTI
6º Laboratório – Exercício 1	Disciplina Projeto de Software – Prof. Enzo Seraphim

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) Salve o diagrama UML no Dia como lab05-1.dia.

7) [Opcional se já feito] Instale o ambiente de desenvolvimento apresentação ecot02-00-instalacoes.pdf em lab 01 de 02/03/2020.

8) **Crie um projeto Maven:**

- Abra a IDE Eclipse Enterprise 2019-12 e crie um novo projeto maven: File | New | Project | Maven | Maven Project; habilitando a opção Create a simple project; Group Id: br.edu.unifei.ecot12; Artifact Id: ecot12-lab05

9) **Defina o compilador 1.8 no projeto Maven:**

- Em Project Explorer abra o arquivo pom.xml e inclua as informações abaixo antes da última linha (</project>):

```
<properties>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
</properties>
```
- Clique com o botão esquerdo do mouse no projeto ecot12-lab05 em Project Explorer e selecione: Maven | Update Project [Ok]

10) **Crie o pacote br.edu.unifei.ecot12.eletrica:**

- Clique com o botão esquerdo do mouse na pasta src/main/java em Project Explorer e selecione New | Package com name br.edu.unifei.ecot12.eletrica

11) **Crie classe abstract ElementoEletrico:**

- Clique com o botão esquerdo do mouse no pacote br.edu.unifei.ecot12.eletrica em Project Explorer e selecione New | Class com Name: ElementoEletrico; habilitando o modificador abstract
- Conforme diagrama UML, declare seu atributo rotulo.
- Clique com o botão esquerdo do mouse após a declaração dos atributos e selecione Source | Generate Getters and Setters, marcando todos os campos da classe.
- Conforme diagrama UML, declare o método abstrato totalElementos

12) **Crie subclasse ComponenteEletrico:**

- Clique com o botão esquerdo do mouse no pacote br.edu.unifei.ecot12.eletrica em Project Explorer e selecione New | Class com Name: ComponenteEletrico; com Superclass: ElementoEletrico; habilitando Inherited abstract method
- Conforme diagrama UML, declare seu atributo tipo.
- Clique com o botão esquerdo do mouse após a declaração dos atributos e selecione Source | Generate Getters and Setters, marcando todos os campos da classe.
- 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 Project Explorer e selecione New | Class com Name: CircuitoEletrico; com Superclass: ElementoEletrico; habilitando Inherited abstract method
- Conforme diagrama UML, declare seus atributos caminho e estado, além do seu relacionamento lista de elementos
- Clique com o botão esquerdo do mouse após a declaração dos atributos e selecione Source | Generate Getters and Setters, marcando todos os campos da classe.

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 Project Explorer e selecione New | Class para criar a classe: App; habilitando public static void main

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.

