

POO - Atividade de Implementação 01

1. Desenvolva uma classe em linguagem de programação C++ denominada `SensorTemperatura` que será utilizada para abstrair um sensor de temperatura industrial interligado a um transmissor de temperatura. O sensor é capaz de medir temperaturas na faixa de 0 a 100 °C, fornecendo uma corrente de saída equivalente que vai de 4 a 20mA. Considere que o objeto da classe `SensorTemperatura` possui um atributo denominado `corrente_saida`, um atributo estático `n_sensores`, que indica quantos sensores de temperatura (número de objetos instanciados) existem em um determinado ambiente físico, um atributo `ligado`, que indica se o equipamento está ligado (`true`) ou desligado (`false`), o atributo `defeito` para representar que o sensor apresenta algum defeito de funcionamento, e o atributo `unidade`, que indica a unidade de engenharia configurada no equipamento (Celsius, Kelvin ou Fahrenheit). Além disto, a classe deve possuir as seguintes características:

- Construtor da classe que recebe como entrada a `unidade` de medição. Considere este atributo como sendo um número inteiro, em que 1 – temperatura em Kelvin, 2 – temperatura em Fahrenheit e qualquer outro valor temperatura em Celsius. O atributo `ligado` deve ser inicializado com o valor `false`, indicando que o equipamento se encontra desligado. O atributo `defeito` deve ser inicializado com o valor `false`, indicando que não há defeitos no sensor;
- Métodos `get` para todos os atributos da classe e método `set` para os atributos `unidade` e `defeito`;
- Métodos `ligar` e `desligar`, que liga (coloca em operação) e desliga (tira de operação), respectivamente, o sensor de temperatura;
- Método `imprimir` que exibe em tela todos os atributos da classe e a temperatura na `unidade` de engenharia configurada no objeto `SensorTemperatura`;
- Métodos estáticos que convertem temperaturas entre unidades. Por exemplo, método `CelsiusToKelvin`, converte um valor informado de temperatura em Celsius para Kelvin;
- Método privado que converte o valor de transmissão entre 4 e 20mA do atributo `corrente_saida` para um valor correspondente de temperatura na `unidade` de engenharia configurada no sensor, retornando este valor;
- Método `efetuaMedicao`, que simula a medição de temperatura efetuada pelo sensor. Inicialmente, deve ser gerado um número entre 0 e 100. Caso este número seja menor do que 5, isto indica que o sensor apresenta algum problema de `funcionamento`. Se o sensor estiver operacional, ele deve efetuar a medição, gerando um número real aleatório entre 4 e 20mA. Este valor deve ser armazenado no atributo `corrente_saida`. O método deve retornar o valor de temperatura na unidade de engenharia configurada no atributo `unidade`.