

## Lista de Exercícios

### Tema 06. Construtores em POO.

#### Recomendações:

- Utilizando a linguagem de programação da sua preferência (Java, Python ou ambas) implemente as classes especificadas nos exercícios a seguir.
- Utilize apropriadamente os modificadores de acesso discutidos na sala de aulas.
- Utilize apropriadamente os métodos Getter e Setter quando necessário. Lembre das validações necessárias associados aos métodos de atualização.
- Implemente o mecanismo de construção da classe segundo as recomendações discutidas em sala de aula.
- Crie em cada caso uma classe cliente, ou aplicação simples onde mostre o uso da classe implementada.

1. Usando como base a classe **Time** apresentada em sala de aulas implemente os seguintes métodos:

- Inclua o construtor da classe
- **boolean isAm()**: retorna verdadeiro se a hora for anterior a 12:00 e falso caso contrário
- **int cron(Time outraHora)**: retorna a quantidade de segundos transcorridos desde a hora enviada como parâmetro, se o parâmetro outraHora for anterior ao tempo do objeto considerar que o parâmetro se refere ao dia seguinte
- **void addSeconds(int secs)**, modifica a hora adicionando a quantidade de segundos enviados como parâmetro, considere que a nova hora pode ser do dia seguinte

2. Usando como base classe **Data** implementada na lista anterior adicione os seguintes métodos:

- Inclua o construtor da classe
- **boolean isPrevious(Data outraData)**: retorna verdadeiro se a data, que passamos por parâmetro, é anterior a data do objeto;
- **int howManyDays(Data outraData)**: retorna a quantidade de dias até a data enviada como parâmetros, se a data for anterior o valor retornado deve ser negativo;

- **String dayOfWeek()**: retorna o dia da semana representado pela data;
3. Usando como base a classe **Contato** desenvolvida na lista anterior, execute o refatoramento da classe incluindo o construtor da mesma e validações para os atributos.
  4. Refatore a classe **Lampada** desenvolvida na lista anterior para incluir o construtor da classe.
  5. Refatore a classe **Ponto2D** desenvolvida na lista anterior para atender os seguintes requisitos:  
Inclua o construtor da classe  
Os pontos são imutáveis uma vez instanciados seus valores não podem ser modificados.
  6. Construa uma classe **Circulo** com os seguintes requisitos:  
Nome da classe: **Circulo**  
Atributos: **x** (abscissa do centro), **y** (ordenada do centro), **r** (raio);  
Métodos:  
    **Constructor**  
    **validaCirculo**  
    **set/get**,  
    **imprimirCirculo()**,  
    **boolean isInnerPoint(Ponto2D)**, retorna verdadeiro se o ponto enviado como parâmetro es interno ao círculo, use como parâmetro um objeto ponto do exercício anterior.  
    **float area()**, retorna a área do círculo  
    **float perimeter()**, retorna o perímetro do círculo  
    **boolean isBiggerThan(Circulo)**, retorna verdadeiro se a área do objeto for maior que a área do círculo enviado como parâmetro.