

## PROGRAMAÇÃO I

### Lista de Exercícios 2

1. Para que é usado tratamento de exceções? Em qual contexto deve-se tratar uma exceção? Em qual contexto deve-se lançar uma exceção?
2. Quais são as restrições impostas pelos modificadores **final** e **static** em atributos e métodos?
3. O que é uma **classe abstrata**?
4. Para que servem as **interfaces**?
5. Quais as vantagens de se utilizar **expressões lambda** na orientação a objetos?
6. Desenvolva uma classe **Estudante**, com os atributos privados **nome** e **três notas**, e os métodos **atribuirPontoExtra** e **calcularMedia**. Instancie a classe Estudante no programa principal e teste os atributos e métodos. Lembre-se de implementar os métodos **getters** e **setters** e fazer os devidos tratamentos de exceção.
7. Altere o método **calcularMedia** do exemplo anterior, para que ele aceite uma **expressão lambda** para definir como o cálculo da média deve ser realizado.
8. Construa classes para representar **Data** e **Hora**. Faça os tratamentos de exceção necessários e implemente alguns métodos. Construa o programa principal que mostre as funcionalidades implementadas. Em ambas as classes faça um método que verifique se a data ou hora está entre duas outras datas ou horas.
9. Crie uma classe chamada **Ingresso** que possui um **valor** em reais e um método **retornaValor**.
  - a. Crie uma classe **IngressoVIP**, que herda de **Ingresso** e possui um valor adicional. Sobrescreva o método **retornaValor**, resultando no valor do ingresso VIP (com o adicional incluído).
  - b. Crie uma classe **IngressoNormal**, que herda de **Ingresso** e possui um método que imprime: "Ingresso Normal".
  - c. Crie uma classe **CamaroteInferior** (que possui a localização do ingresso e métodos para acessar e imprimir esta localização) e uma classe **CamaroteSuperior**, que é mais cara (possui valor adicional). Esta última também deve sobrescrever o método **retornaValor**, para calcular o valor do ingresso. Ambas as classes herdam da classe **IngressoVIP**.
  - d. Faça os devidos tratamentos de exceção em todas as classes desenvolvidas.
10. Construa uma família de classes (com classes base e subclasses) que represente um **Movel** (móveis de uma casa). Utilize herança e faça os tratamentos de exceção necessários. Construa o programa principal que mostre as funcionalidades implementadas.