

**Faculdade Dom Bosco de Porto Alegre**  
**Sistemas de Informação**  
**Estrutura de Dados**  
**Profª Adriana Scherer**

**Exercícios - Introdução OO – Encapsulamento e Polimorfismo**

1. Crie uma classe de nome Triangulo que possui os atributos: base e altura. E o método:  
Área: que retorna o cálculo da área do triângulo. Sabe-se que o cálculo da área do triângulo é:  $\text{area} = (\text{base} * \text{altura}) / 2$ .  
Após, implemente em java um programa que lê os valores da base e da altura e mostre o valor da área.

1. Altere o exercício anterior e defina os atributos da classe Triangulo como private.
2. Crie uma classe funcionário que tenha o nome (String) e e-mail (String) como atributos privados e defina um método privado que gere o endereço de e-mail que deve ser a primeira letra do pré-nome, seguido dos demais nomes e sobrenome, concatenando com a string @estruturadados.com.br.
3. Crie uma classe para instanciar um objeto funcionário, solicitar o nome e após mostrar o nome acompanhado do e-mail. Exemplo: José Silva, [jsilva@estruturadados.com.br](mailto:jsilva@estruturadados.com.br)
4. Criar uma classe que deverá ser a modelagem de uma calculadora que efetua somente as quatro operações básicas e possui os seguintes atributos privados: num1 (double) e num2 (double). Sendo os atributos privados, defina um método set e um método get para cada atributo. Além dos métodos set e get, deverão ser definidos os seguintes métodos:
  - somaNumeros() deve retornar a soma do num1 + num2;
  - subtrNumeros() deve retornar o resultado da subtração de num1 – num2;
  - multNumeros() deve retornar o produto de num1 \* num2;
  - divNumeros() deve retornar o resultado da divisão de num1 / num2.

Como complemento do exercício, crie uma classe que deve possuir o método main onde:

- será instanciado um objeto calculadora;
- os valores para os atributos (num1 e num2) serão lidos via teclado;
- deverá ser definida uma variável inteira chamada operacao;
- leia através do teclado um valor para a variável operacao (valor de 1 a 4);
- exibir o resultado da operação escolhida, com mensagem adequada.

5. Crie uma classe para simular o funcionamento de um motor. A classe Motor deve possuir os seguintes atributos privados: velocidade (inteiro) em RPM (rotações por minuto), fabricante (String), voltagem (inteiro), status (String) do motor (ligado/desligado) e os métodos: liga motor, desliga motor, verifica velocidade do motor. O construtor da classe deve atribuir valores para os atributos fabricante, voltagem e velocidade.
6. Crie uma classe que instancia três objetos (motor0, motor1 e motor2) da classe motor e imprima na tela os valores dos atributos: fabricante, voltagem e velocidade. Não se esqueça de definir o status do motor!
7. Crie uma classe que estenda as características do exercício 5 acrescentando dois métodos: um para aumentar a velocidade e outro para reduzir a velocidade. A velocidade não pode passar de 5000 RPM e não pode ser menor que 0.
8. Altere o exercício 6 e:
  - a) Aumente a velocidade do motor0 em 10 RPM
  - b) Aumente a velocidade do motor1 em 6 RPM
  - c) Diminua a velocidade do motor2 até que ele pare
  - d) Exibir o status do motor2.