

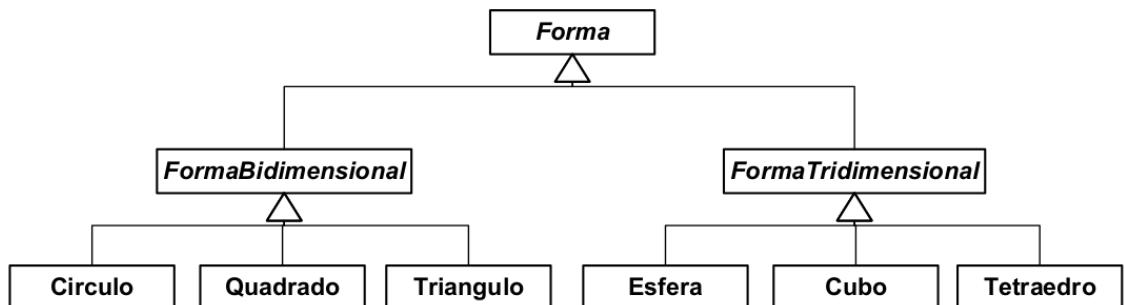


## Lista de Exercícios 05 – Polimorfismo e Exceções

### Instruções

- Todos os exercícios que envolvem programas devem ser resolvidos por programas em linguagem C++;
- Na solução dos exercícios, devem ser utilizados os conceitos listados no cabeçalho desta lista;
- Eventuais dúvidas podem ser sanadas com o professor.

1. Implemente a hierarquia Formato como descrito pelo diagrama abaixo e por este texto:
  - a. Cada FormaBidimensional deve conter um método getArea() que calcula sua área;
  - b. Cada FormaTridimensional deve conter os métodos getArea() e getVolume(), para calcular área e volume, respectivamente;
  - c. Cada classe deve possuir um destrutor específico, que imprima o nome da própria classe;
  - d. Crie um driver com um vetor de ponteiros para Forma, cada ponteiro deve apontar para um dos objetos de cada classe concreta da hierarquia;
  - e. O driver deve imprimir a classe de origem do objeto apontado por cada ponteiro, o que deve ser determinado em tempo de execução;
  - f. O driver deve imprimir, para cada objeto apontado, se trata-se de uma FormaBidimensional ou FormaTridimensional, o que deve ser determinado em tempo de execução
    - i. Se a forma for bidimensional, imprima sua área;
    - ii. Se a forma for tridimensional, imprima sua área e volume.
  - g. Processando os objetos polimorficamente, o destrutor deve ser invocado através do método delete().



2. Implemente as classes e o relacionamento entre classes descritos no diagrama UML abaixo. Note que apenas as classes do último nível da hierarquia são concretas. No driver, crie um vetor de ponteiros para classe Laurasiatheria, em cada posição, adicione ponteiros para cada uma das classes concretas, em ordem alfabética.

- a. Utilizando o comportamento polimórfico, utilize os ponteiros criados para invocar o método Som() adequado para cada classe apontada;
- b. Determine, em tempo de execução, a classe de cada objeto apontado pelo vetor;

