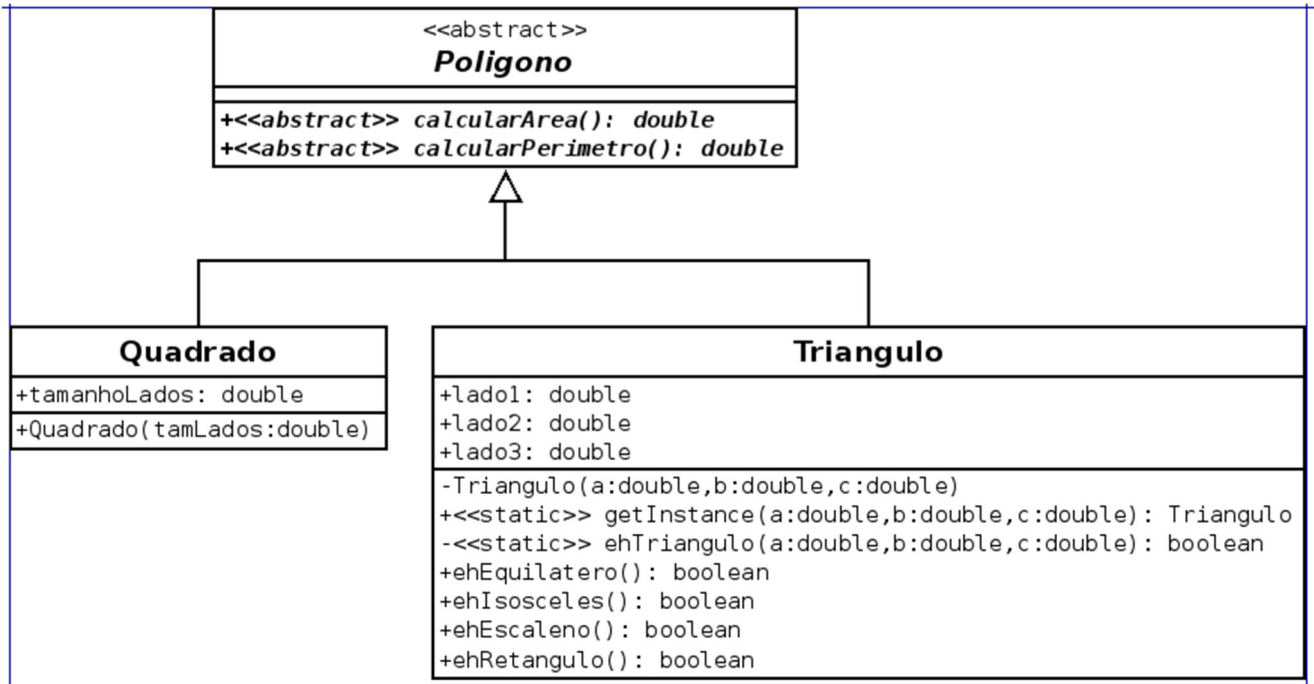


## Aula 4 – Lista de exercícios



1. Dado o Diagrama de Classes acima, crie um projeto em Java (no NetBeans), chamado `exercicio1`, e:
  - a. Crie a classe abstrata **Poligono**, com os seus métodos abstratos `calcularArea` e `calcularPerimetro`;
  - b. Crie a classe **Quadrado** (que estende **Poligono**), com seu atributo `tamLados` e implemente seus métodos: o construtor constante do diagrama, a implementação dos métodos abstratos previstos em **Poligono**;
  - c. Crie a classe **Triangulo**, com seus atributos (conforme modelo) e implemente seus métodos: **(alguns já foram resolvidos nos exercícios da aula anterior)**
    - i. `ehEquilatero()`, que deve retornar verdadeiro se o objeto criado representar um triângulo equilátero, e falso caso contrário (é equilátero se tiver os 3 lados iguais);
    - ii. `ehIsosceles()`, que deve retornar verdadeiro se o objeto criado representar um triângulo isósceles, e falso caso contrário (é isosceles se pelo menos dois lados tiverem o mesmo tamanho);
    - iii. `ehEscaleno()`, que deve retornar verdadeiro se o objeto criado representar um triângulo escaleno, e falso caso contrário (é escaleno se os 3 lados forem de tamanhos diferentes);
    - iv. `ehRetangulo()`, que deve retornar verdadeiro se o objeto criado representar um triângulo retângulo, e falso caso contrário (é retângulo se não for equilátero e se um dos lados for hipotenusa);  **$hip^2 = catAdj^2 + catOpo^2$**
    - v. `ehTriangulo` (privado estático), que deve receber os lados como parâmetros e retornar verdadeiro se os lados formarem um triângulo, e falso caso contrário (só é triângulo se cada lado for diferente de 0 e menor que a soma dos outros dois);
    - vi. construtor (privado), para atribuir os lados (a, b, c) aos respectivos atributos;
    - vii. O método estático (static) `getInstance`, que pode ser chamado sem a instância prévia da classe. O método receberá os três lados e chamará o método `ehTriangulo`, passando-os como parâmetros; se o método `ehTriangulo` retornar `true`, o `getInstance` deve instanciar um **Triangulo** e retorná-lo; se `ehTriangulo` retornar `false`, o `getInstance` deve retornar `null`;
    - viii. Os métodos abstratos previstos em **Poligono**.

Obs.: para calcular a área de um triângulo, dados os seus três lados, pode-se utilizar o Teorema de Heron (semi-perímetro):

$$A = \sqrt{p \cdot (p - a) \cdot (p - b) \cdot (p - c)}$$

Onde:  $p = \frac{(a + b + c)}{2}$