Programação Orientada a Objetos

Prof. Delano M. Beder

Desafio – Polígonos

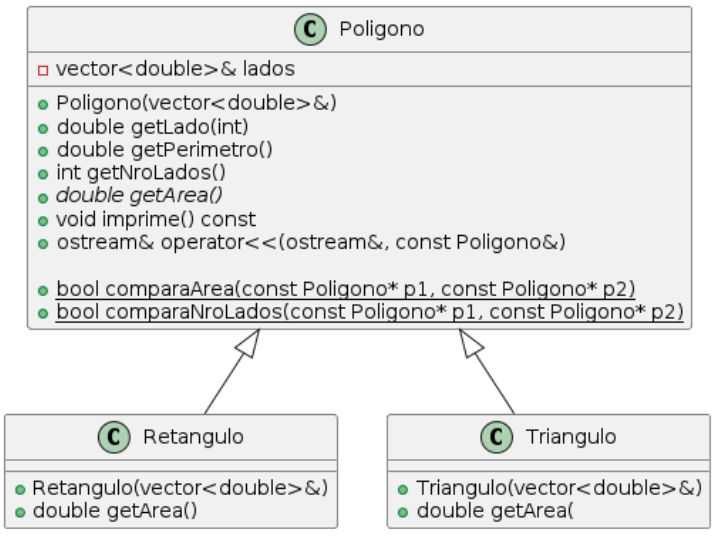
Poligono.h

```
#ifndef POLIGONO_H
#define POLIGONO_H

#include <iostream>
#include <vector>
using namespace std;

class Poligono {
public:
  Poligono(vector<double>&);
  virtual ~Poligono();
  double getLado(int) const;
  double getPerimetro() const;
  int getNroLados() const;
  virtual double getArea() const = 0;
  virtual void imprime() const;
  friend ostream& operator<<(ostream&, const Poligono&);

  static bool comparaArea(const Poligono* p1, const Poligono* p2);
  static bool comparaNroLados(const Poligono* p1, const Poligono* p2);
private:
  vector<double>& lados;
};

#endif /* POLIGONO_H */
```

Desafios:

1. Poligono.cpp (com as implementações dos métodos da classe Poligono)
2. Implemente a classe Retangulo (Retangulo.h e Retangulo.cpp)
   Área do Retângulo => (base x altura)/2
3. Implemente a classe Triangulo (Triangulo.h e Triangulo.cpp)

$$A = \sqrt{p.(p-a).(p-b).(p-c)} \qquad p = \frac{a+b+c}{2}$$



4. Teste suas implementações com o main.cpp abaixo:

```
#include <iostream>
#include <algorithm> // std::sort
#include <vector>   // std::vector
#include "Poligono.h"
#include "Triangulo.h"
#include "Retangulo.h"

int main() {

  vector<Poligono *> poligonos;

  vector<double> v1{3, 4, 5};
  poligonos.push_back(new Triangulo(v1));

  vector<double> v2{3, 4, 3, 4};
  poligonos.push_back(new Retangulo(v2));

  vector<double> v3{3, 3, 3};
  poligonos.push_back(new Triangulo(v3));

  vector<double> v4{2, 3, 2, 3};
  poligonos.push_back(new Retangulo(v4));

  cout << "poligonos:" << endl;

  for (unsigned long int i = 0; i < poligonos.size(); i++) {
    cout << *poligonos[i] << endl;
  }

  cout << endl << "poligonos (ordenado pela Area):" << endl;

  sort(poligonos.begin(), poligonos.end(), Poligono::comparaArea);

  for (unsigned long int i = 0; i < poligonos.size(); i++) {
    poligonos[i]->imprime();
  }

  cout << endl << "poligonos (ordenado pelo Nro Lados):" << endl;

  sort(poligonos.begin(), poligonos.end(), Poligono::comparaNroLados);

  for (unsigned long int i = 0; i < poligonos.size(); i++) {
    poligonos[i]->imprime();
  }
  return 0;
}
```