



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE - CTS
DEPARTAMENTO DE COMPUTAÇÃO – DEC

DISCIPLINA: LINGUAGENS DE PROGRAMAÇÃO 2
PROFESSOR ANTONIO CARLOS SOBIERANSKI
a.sobieranski@ufsc.br

ENUNCIADO TRABALHO T5 – Shapes distintos em uma lista genérica

A ser desenvolvido em: DUPLAS

O T5, em continuidade com a aula sobre Construtores / Destrutores, será a implementação de algumas classes de formas 2D e 3D geométricas distintas. O armazenamento dos objetos após instanciados ocorrerá em estrutura de dados genérica que permita tal armazenamento.

Para esta implementação, considerar as seguintes classes abaixo:

Circle: possui como atributo somente *m_ratio*;

Square: possui como atributo somente *m_edge*;

Triangle: possui como atributos 3 vértices bidimensionais (*m_p1*, *m_p2*, *m_p3*) e 3 *Edges* (lados) (*m_e1*, *m_e2*, *m_e3*). Os triângulos podem ser regulares ou irregulares (triângulos quaisquer). Podem ser inicializados via construtores tanto por vértices como por lados.

Sphere: possui como atributo *m_ratio*;

Tetrahedron: possui como atributos 4 vértices tridimensionais (*m_p1*, *m_p2*, *m_p3*, *m_p4*). Podem ser regulares ou irregulares (tetraedrons quaisquer).

Cube: possui como atributo somente *m_edge*.

Todas as classes acima devem possuir um método que calcula a sua área, lembrando que no caso das formas 3D a área é a soma das superfícies ou facetas que as compõem. O armazenamento deve ocorrer em um **vector** `< GenericShape >`, onde **GenericShape** é uma classe especializada no armazenamento das formas acima, através de vários **construtores** especializados (um para cada classe acima).

O programa será alimentado com o arquivo de texto contendo diversos objetos armazenados, com seus atributos pré-estabelecidos, que será solicitado ao início da execução do programa:

Exemplo:

```
if(NDEBUG) cout << "Enter with a shape configuration file to load:";  
string path;  
cin >> path;
```

Abaixo, exemplo de input e a respectiva saída do programa:

input1.txt	output
circle 10 square 10 triangle 1 1 1.4142 triangle 208 203 137	Object #0 is circle -> 314.16 Object #1 is square -> 100.00 Object #2 is triangle -> 0.50 Object #3 is triangle -> 13262.85 Object #4 is sphere -> 1256.64 Object #5 is cube -> 600.00 Object #6 is tetrahedron -> 2.37 Object #7 is triangle -> 0.50 Object #8 is triangle -> 0.87 Object #9 is tetrahedron -> 110.85

sphere 10 cube 10 tetrahedron 0,0,0 0,1,0 1,0,0 0,0,1 triangle 0,0 1,0 1,1 triangle 1.4142 1.4142 1.4142 tetrahedron 0,0,0 8,0,0 4,6.9282,0 4,2.309,6.5319	
---	--

Obs.:

1. Todas as classes possuem construtor *default*, porém, não serão utilizados. Ou seja, se o *input* especifica um círculo, o mesmo será sempre construído por parâmetros.
2. A única classe que possuirá mais de uma opção de construtor (além do *default* mencionado acima), será a classe triângulo (inicializa por lados ou coordenadas bi-dimensionais). Todas as demais classes possuirão exatamente 1 construtor com parâmetros.
3. Triângulos podem ser inicializados por 3 lados com quebra de linha no arquivo, ou por 3 coordenadas bi-dimensionais, separadas por vírgula e quebra de linha (ver *input1.txt*).
4. Tetraedros inicializam com 4 coordenadas tridimensionais, separadas por vírgula e quebra de linha, respectivamente (ver *input1.txt*)
5. Usar o armazenamento em uma estrutura genérica, com construtores especializados, chamando uma função que retorna a área de cada um dos objetos inseridos.

Dicas:

- a) O penúltimo tetraedro é uma composição de 3 triângulos retângulos com lados de tamanho 1.0 com hipotenusa de 1.4142, e mais 1 triângulo com 3 lados de 1.4142 (fica fácil visualizar, já que inicia na origem 0,0).
- b) O último tetraedro possui configurações de coordenadas que formam todos os lados de mesmo tamanho (tamanho = 8).
- c) A menor distância entre 2 pontos é uma linha reta, logo utilizem distância Euclideana para tal, que não deixa de ser uma generalização de Pitágoras (a própria hipotenusa).
- d) A área de triângulos quaisquer pode ser obtida pelo teorema de Heron.
- e) Tetraedros são composições de triângulos no espaço 3D, logo a área (faceta) pode ser também obtida pelo teorema de Héron (basta ter os lados).