



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](mailto:a.sobieranski@ufsc.br)

## ENUNCIADO TRABALHO T2 – Classe POLYGON

### A ser desenvolvido em Duplas (Data de Entrega: 05/05/2023)

Realizamos em sala o exercício da Classe Polígono nos dias 17 e 19 de Abril, conforme consta no Moodle.

Pede-se:

Utilizando especificamente o código que consta em : **19.04.23 - Classe Polygon - Calculando perímetro**, modificar o mesmo para que todo e qualquer atributo da Classe **Polygon** e da Classe **Point2D** sejam dinâmicos.

Ou seja (exemplo somente):

```
private: //(Polygon.hpp)
vector< Point2D > m_minhaListaDeVertices;    →    Point2D *m_minhaListaDeVertices;
```

```
private: //(Point2D.hpp)
int m_x, m_y;                                →    int *m_x, *m_y;
```

#### Requisitos Importantes:

- A entrada do programa será um arquivo texto, contendo a quantidade de Pontos 2d e a sequência de coordenadas x e y (do tipo inteiro) que compõem o polígono (vide exemplo). Deverá ser implementada a leitura deste arquivo no **main.cpp** e passado para a classe *Polygon* via método de acesso **E** construtor (as 2 opções devem ser possíveis, e escolher 1 na sua implementação). Ou seja, **o main**, através de um *cout* e *cin*, pedirá ao usuário qual o nome do arquivo a ser lido. As instâncias da classe *Polygon* devem somente receber a lista de pontos, e nunca ler o arquivo diretamente.
- Respeitar o encapsulamento, e implementar métodos de acesso de forma apropriada;
- Alocar apropriadamente a memória para os atributos e limpar memória quando necessários (*new*, *delete*). Importante não ter *memory-leak* no código;
- Como todos os atributos são dinâmicos, não esquecer de atribuí-los a NULL nos construtores *default* (que não inicializarem com parâmetros);
- Não será permitido uso de **std::vector**, **std::pair** ou **std::tuple** em nenhuma parte do programa;
- A saída do programa será (i) a área e (ii) o perímetro do polígono informado via arquivo;

#### Entregável no Moodle:

- Código fonte, considerando somente *cpp*'s e *hpp*'s (ou *.h*) do programa desenvolvido. Deve compilar e ser possível de testes.
- Código fonte não compilou: ZERO (compilarei com "**g++ \*.cpp -o exec**")

#### Dicas:

Alguém disse um dia: "Debugar salva vidas...."

Para testar *Memory-leak*, pesquisar como fazer na sua IDE.