PROGRAMACIÓN ORIENTADA A OBJETOS

GRADO EN INGENIERÍA INFORMÁTICA
SEGUNDO CURSO

DEPARTAMENTO DE INFORMÁTICA Y ANÁLISIS NUMÉRICO
ESCUELA POLITÉCNICA SUPERIOR
UNIVERSIDAD DE CÓRDOBA

CURSO ACADÉMICO: 2012 – 2013

- **Número de práctica: 6**

- **Objetivo**
  - Definir una **clase abstracta** y funciones **virtuales**
  - Utilizar el contenedor STL **vector**

- **Definición de la clase abstracta Poligono2D**
  - **Descripción**
    - Poligono2D será una clase abstracta que tendrá dos métodos virtuales: calcularPerimetro y calcularArea.
    - A partir de la clase Poligono2D, se definirán las clases Triangulo2D y Cuadrado2D.

  - **Atributos**
    - _vertices: atributo del tipo contenedor STL **vector** de objetos de tipo **Punto2D**
    - Observación**:**
      - Para usar el contenedor STL vector, el fichero poligono2D.hpp deberá incluir la sentencia
        #include <vector>

  - **Funciones o métodos públicos**
    - **Constructor**:
      - Recibe como parámetro el número de vértices del polígono.
      - Se deberá usar el método **"resize"** del contenedor STL vector
    - **getNumeroVertices**:
      - Devuelve el número de vértices del Poligono2D
      - Se deberá usar el método **"size"** del contenedor STL vector
    - **getVertice**:
      - Recibe como parámetro un índice y devuelve una referencia al Punto2D del Poligono2D que ocupa el lugar indicado por el índice.
    - **setVertice**:

        - Recibe como parámetro un índice y un Punto2D “p”
        - Asigna el valor del Punto2D “p” al vértice del Poligono2D que ocupa el lugar indicado por el índice.
    - **calcularPerimetro**:
        - Función virtual
        - Se definirá en las clase herederas
    - **calcularArea**
        - Función virtual
        - Se definirá en las clase herederas

- **Definición de la clase abstracta Triangulo2D**
    - **Descripción**
        - Triangulo2D es una clase que hereda de forma pública de la clase abstracta Poligono2D

    - **Atributos**
        - No tiene atributos propios
        - Hereda el atributo de la clase abstracta Poligono2D, pero con un tamaño de **tres** vértices

    - **Funciones o métodos públicos**
        - **Constructor**:
            - Crea un Poligono2D con tres vértices
            - Se recomienda usar el iniciador de la clase base Poligono2D
        - **esTriangulo**
            - Comprueba que los vértices no están alineados, es decir, cumplen la **propiedad triángula**r
                $|b - c| < a < |b + c|$
            - donde a, b y c son las longitudes de los lados del triángulo
            - La función devuelve “true” si los vértices forman un triángulo; “false”, en caso contrario.
        - **calcularPerimetro**:
            - Calcula el perímetro de un Triangulo2D
        - **calcularArea**
            - Calcula el área de un triángulo utilizando la fórmula de **Herón de Alejandría**
                $$area = \sqrt{s(s-a)(s-b)(s-c)}$$
                - donde
                    - a, b y c son las longitudes de los lados del triángulo
                    - s = semiperímetro del triángulo
            - Observación:
                - La precondición {s (s – a) (s – b) (s – c) > 0} se comprobará utilizando el método **esTriángulo**, que garantiza que los vértices forman realmente un triángulo.
                - Si lo vértices no forman un triángulo, el valor devuelto será 0.0.

- **Definición de la clase abstracta Cuadrado2D**
  - **Descripción**
    - Cuadrado2D es una clase que hereda de forma pública de la clase abstracta Poligono2D

  - **Atributos**
    - No tiene atributos propios
    - Hereda el atributo de la clase abstracta Poligono2D, pero con un tamaño de cuatro vértices

  - **Funciones o métodos públicos**
    - **Constructor**:
      - Crea un Poligono2D con **cuatro** vértices
      - Se recomienda usar el iniciador de la clase base Poligono2D

    - **esCuadrado**
      - Comprueba que los vértices forman realmente un cuadrado.
      - La función devuelve "true" si los vértices forman un cuadrado; "false", en caso contrario.

    - **calcularPerimetro**:
      - Calcula el perímetro de un Cuadrado2D

    - **calcularArea**
      - Calcula el área de un cuadrado
      - Observación:
        - Se utilizará previamente el método esCuadrado para comprobar que los vértices forman realmente un cuadrado.
        - Si los vértices no forman un cuadrado entonces el valor devuelto será 0.0

- **Observación:**
  - Se deben codificar los siguientes ficheros
    - **poligono2D.hpp**
      - Contiene la declaración y la definición de la clase abstracta Poligono2D
    - **triangulo.hpp**
      - Contiene la declaración de la clase Triangulo2D
    - **triangulo.cpp**
      - Contiene la definición de los métodos de la clase Triangulo2D
    - **cuadrado2D.hpp**
      - Contiene la declaración de la clase Cuadrado2D
    - **cuadrado2D.cpp**
      - Contiene la definición de los métodos de la clase Cuadrado2D
  - Además, se deberán utilizar los ficheros de la clase Punto2D codificados en la práctica nº 3

    - **punto2D.hpp**
    - **punto2D.cpp**
  - Las clase codificadas deberán permitir la ejecución del programa de ejemplo contenido en el fichero **practica_6.cpp**
  - Se deberán utilizar el espacio de nombres **poo**
  - Se deberá crear un fichero **makefile** de compilación