PROGRAMACIÓN ORIENTADA A OBJETOS

GRADO EN INGENIERÍA INFORMÁTICA
SEGUNDO CURSO

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

CURSO ACADÉMICO: 2011 – 2012

- **Examen de prácticas: 3 de septiembre de 2012**

- **Objetivo**
  - Definir y utilizar siguientes clases
    - **Empresa**:
      - Representa una empresa comercial
    - **Pyme**:
      - Hereda de forma pública de Empresa
      - Representa a una pequeña y mediana empresa que posee un propietario.
    - **Apyme**:
      - Clase que representa el TAD “asociación de pequeñas y medianas empresas” y está compuesta por “pymes”

- **Definición de la clase Empresa**

  - **Posee los atributos**
    - _nombre
    - _dirección
    - _CIF

  - **Funciones o métodos públicos**
    - Constructor:
      - Versión 1
        - Constructor parametrizado
        - Todos los argumentos deberán tener un valor por defecto
      - Versión 2
        - Constructor de copia

  - **Funciones de acceso**
    - getNombre

        - getDireccion
        - getCIF

    - **Funciones de modificación**
        - setNombre
        - setDireccion
        - setCIF

    - **Funciones de lectura y escritura**
        - leerEmpresa
            - Lee desde el teclado los valores de los atributos de la Empresa actual.
        - escribirEmpresa
            - Escribe por pantalla los valores de los atributos de la Empresa actual.

- **Definición de la clase Pyme**

    - **Descripción**
        - Pyme hereda de forma pública de la clase Empresa

    - **Atributo propio**
        - _propietario:
            - nombre de la persona responsable de la pequeña y mediana empresa

    - **Funciones o métodos públicos**
        - Constructor:
            - Versión 1
                - Constructor parametrizado
                - Todos los argumentos deberán tener un valor por defecto
            - Versión 2
                - Constructor de copia

        - **Función de acceso**
            - getPropietario

        - **Funciones de modificación**
            - setPropietario
            - Sobrecarga del operador de asignación “=”

        - **Funciones de lectura y escritura**
            - leerPyme
                - Lee desde el teclado los valores de los atributos de la Pyme actual.
            - escribirPyme
                - Escribe por pantalla los valores de los atributos de la Pyme

actual.

- **Sobrecarga de los operadores “<<” y “>>”**
  - Sobrecargar el operador “>>” para leer desde el flujo de entrada actual los valores de los atributos de la Pyme actual.
  - Sobrecargar el operador “<<” para escribir en el flujo de salida actual los valores de los atributos de la Pyme actual.

- **Definición de la clase Apyme**

  - **Descripción**
    - Clase que representa el TAD “asociación de pequeñas y medianas empresas” y está compuesta por “pymes”

  - **Atributos propios**
    - _nombreApyme
    - _vectorPymes
      - Se podrá decidir cómo implementar el vector de Pymes:
        - mediante memoria dinámica
        - o usando el contenedor STL vector

  - **Funciones o métodos públicos**
    - Constructor: Apyme
      - Constructor sin argumentos

  - **Destructor de la clase Apyme**
    - Libera la memoria ocupada por la Apyme actual

  - **Métodos de acceso para los atributos propios**
    - getNombreApyme
      getVectorPymes
    - getNumeroApymes

  - **Otras funciones de consulta**
    - getPyme:
      - Recibe como parámetro un “indice” de tipo entero
      - Devuelve una referencia a la Pyme que ocupa el lugar señalado por “indice”.
    - Sobrecarga del operador []:
      - Recibe como parámetro un “indice” de tipo entero
      - Devuelve una referencia a la Pyme que ocupa el lugar señalado por “indice”.

  - **Métodos de modificación para los atributos propios**
    - setNombreApyme
    - setVectorPymes

- **Sobrecarga del operador “=”**

- **Funciones de lectura o escritura**
  - escribirApyme
    - Escribe por pantalla el nombre y número de Pymes de la Apyme y los datos de sus Pymes.
  - leerApyme
    - Lee del teclado el nombre y número de Pymes de la Apyme y los datos de sus Pymes

  - grabarApymeEnFichero:
    - Recibe como parámetro el nombre de un fichero
    - Escribe en el fichero los datos de la Apyme:
      - La primera línea del fichero debe contener el nombre de la Apyme
      - La segunda línea debe contener el número de Pymes
      - Las demás líneas deben contener los datos de cada Pyme

---

- **Observaciones**
  - Las clases definidas deberán permitir la ejecución de los tres programas de prueba:
    - test1.cpp: comprobación de la clase Empresa
    - test2.cpp: comprobación de la clase Pyme
    - test3.cpp: comprobación de la clase Apyme

  - Todos los ficheros del examen se crearán en un único directorio.
  - Se pueden añadir los atributos y métodos privados que se consideren necesarios, además de los indicados en la especificación de cada ejercicio.

  - Nota de ayuda:
    - Si no sabe hacer alguna de las funciones, al menos, declárela y déjela sin código para que el programa pueda compilar.
    - Se evaluará la parte del programa que funcione.

  - Se valorará
    - La utilización de un espacio de nombres
    - La claridad del código
    - La documentación utilizando doxygen

- **Nota**
  - Si se desea leer desde el teclado una cadena (string) que contenga “espacios en blanco” entonces se debe usar el método ***getline***

- Por ejemplo:

```
// cadena.cpp
// Permite leer una frase con varias palabras y espacios en blanco

#include <iostream>
using namespace std;

int main()
{
   string frase;

   cout << "Introduce una frase con varias palabras: ";

   getline(cin,frase);

   cout << "Tu frase es: " << frase << endl;
}
```

- Si se desea leer desde un “flujo de entrada” entonces

```
   getline(flujo,frase);
```