



# PROGRAMACIÓN ORIENTADA A OBJETOS

GRADO EN INGENIERÍA INFORMÁTICA  
SEGUNDO CURSO



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

CURSO ACADÉMICO: 2011 - 2012

- 
- **Examen de prácticas: 23 de enero de 2012**
  - **Objetivo**
    - Definir y utilizar siguientes clases
      - **Persona:**
      - **Socio:**
        - Hereda de forma pública de Persona
        - Representa a una persona que es socia de una ONG
      - **Ong:**
        - Clase que representa el TAD “organización no gubernamental” y está compuesta por socios

- 
- **Definición de la clase Persona**
    - Posee los atributos
      - `_nombre`
      - `_apellidos`
      - `_edad`
    - 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`
      - `getApellidos`
      - `getEdad`

- Funciones de modificación
    - setNombre
    - setApellidos
    - setEdad
  - Funciones de lectura y escritura
    - leerPersona
      - Lee desde el teclado los valores de los atributos de la Persona actual.
    - escribirPersona
      - Escribe por pantalla los valores de los atributos de la Persona actual.
- 

## • Definición de la clase Socio

- Socio hereda de forma pública de la clase Persona
- Atributo propio
  - \_donativo, que representa el dinero que dona mensualmente a la ONG
- 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
    - getDonativo
  - Funciones de modificación
    - setDonativo
    - Sobrecarga del operador de asignación “=”
  - Funciones de lectura y escritura
    - leerSocio
      - Lee desde el teclado los valores de los atributos del Socio actual.
    - escribirSocio
      - Escribe por pantalla los valores de los atributos del Socio actual.
  - Sobrecarga de los operadores “<<” y “>>”
    - Sobrecargar el operador “>>” para leer desde el teclado los

- valores de los atributos del Socio actual.
  - Sobrecargar el operador “<<” para escribir por pantalla los valores de los atributos del Socio actual.
- 

- **Definición de la clase Ong**

- Descripción
  - Clase que representa a las organizaciones no gubernamentales.
- Atributos
  - \_nombreOng
  - \_vectorSocios
    - Cada alumno deberá decidir cómo implementar el vector de socios: mediante memoria dinámica o usando el contenedor STL vector
- Funciones o métodos públicos
  - Constructor: Ong
    - Versión 1: constructor sin argumentos
- Destructor de la clase Ong
  - Libera la memoria ocupada por la Ong actual
- Métodos de acceso para los atributos propios
  - getNombreOng
  - getVectorSocios
  - getNumeroSocios
- Otra función de consulta
  - getSocio:
    - Recibe como parámetro un “índice” de tipo entero
    - Devuelve una referencia al Socio que ocupa el lugar señalado por “índice”.
  - Sobrecarga del operador []:
    - Recibe como parámetro un “índice” de tipo entero
    - Devuelve una referencia al Socio que ocupa el lugar señalado por “índice”.
- Métodos de modificación para los atributos propios
  - setNombreOng
  - setVectorSocios
- Sobrecarga del operador “=”
- Funciones de lectura o escritura
  - escribirOng

- Escribe por pantalla el nombre y numero de socios de la Ong y los datos de sus socios.
- leerOng
  - Lee del teclado el nombre y numero de socios de la Ong y los datos de sus socios
- grabarOngEnFichero:
  - Recibe como parámetro el nombre de un fichero
  - Escribe en el fichero los datos de la Ong:
    - La primera línea del fichero debe contener el nombre de la Ong
    - La segunda línea debe contener el número de Socios
    - Las demás líneas deben contener los datos de cada Socio

- **Observaciones**

- Las clases definidas deberán permitir la ejecución de los tres programas de prueba:
  - test1.cpp: comprobación de la clase Persona
  - test2.cpp: comprobación de la clase Socio
  - test3.cpp: comprobación de la clase Ong
- Todos los ficheros del examen se crearán en un único directorio.
- Puede añadir los atributos y métodos privados que considere 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