



FACULTAD DE INFORMÁTICA

Más sobre clases y objetos

TALLER

Programación orientada a objetos — Unidad 2

Autor: Luis Hernández Yáñez

FdI
UCM

Cuestiones

¿Por qué? ¿Cómo? ¿Cuándo? (*resolución individual*)

- ✓ ¿Por qué no es correcto intentar inicializar los atributos en el mismo momento en el que se declaran?
Es decir, ¿por qué no es correcto algo como lo siguiente?

```
class Punto {  
...  
private:  
    double _x = 0, _y = 0;  
};
```
- ✓ ¿Por qué es más adecuado colocar el código de inicialización en un constructor que en una función miembro como `inicializa()`?
- ✓ Explica la importancia de los métodos que la Forma canónica ortodoxa señala como esenciales.

Programación orientada a objetos

Unidad 2 – Taller – Página 1

FdI
UCM

En construcción

¿Qué aparecerá en la pantalla? (*resolución individual*)

Indica qué mostrará en la pantalla el programa que sigue (sin ejecutarlo):

```
#include <iostream>  
using namespace std;  
  
class Una {  
public:  
    Una(int d = 1) : _dato(d) { }  
    void mostrar() const { cout << _dato; }  
private:  
    int _dato;  
};
```

continúa

Programación orientada a objetos

Unidad 2 – Taller – Página 2

FdI
UCM

En construcción

```
class Dos {  
public:  
    Dos(int d = 2) : _dato(d), _obj1(d) { }  
    void mostrar() const { cout << "(" << _dato << ",";  
        _obj1.mostrar(); cout << ")"; }  
private:  
    int _dato;  
    Una _obj1;  
};  
  
class Tres {  
public:  
    Tres(int d = 3) : _obj1(d) { }  
    void mostrar() const { cout << "("; _obj1.mostrar();  
        cout << ","; _obj2.mostrar(); cout << ")"; }  
private:  
    Una _obj1;  
    Dos _obj2;  
};
```

continúa

Programación orientada a objetos

Unidad 2 – Taller – Página 3

```
int main() {
    Tres unObjeto;
    unObjeto.mostrar(); cout << endl;
    Tres otroObjeto(5);
    otroObjeto.mostrar(); cout << endl;
    Dos objeto2(9);
    objeto2.mostrar(); cout << endl;

    return 0;
}
```

Una clase más completa (*resolución en grupo*)

A partir de la clase **Contador2** del taller de la semana anterior, hay que crear otra clase, **Contador3**, que sea igual que la anterior y que se ajuste a la Forma canónica ortodoxa. Además, se han de declarar como constantes los métodos para los que resulte procedente. Y se ha de probar exhaustivamente la nueva clase en una función **main()**.

Mejorando la clase (*resolución en grupo*)

Se ha de modificar la clase **Empleado** del taller de la semana anterior, de forma que se ajuste a la Forma canónica ortodoxa. Además, se han de declarar como constantes los métodos para los que resulte procedente. Y se ha de probar exhaustivamente la clase en una función **main()**.