



FACULTAD DE INFORMÁTICA

Ejercicios de repaso

TALLER

Programación orientada a objetos — Unidad 5

Autor: Luis Hernández Yáñez

FdI
UCM

Resolución de los ejercicios

Se dispone de las clases **Persona**, **Fecha** y **Hora**.

Se deben utilizar siempre que resulte adecuado.

Cada ejercicio consiste en el desarrollo de una clase y un programa de prueba para la misma.

Las clases han de estar protegidas frente a inclusiones múltiples y ser módulos de biblioteca (cabecera .h e implementación .cpp).

Todos los métodos han de ser implementados fuera de la estructura **class**.

En las resoluciones se aplicará todo lo visto hasta ahora.

Programación orientada a objetos

Unidad 5 – Taller – Página 1

FdI
UCM

Ejercicio nº 1

Crea una clase **vehículo** que modele la información que se mantiene en un taller sobre cada vehículo que reparan: modelo, propietario (una persona), importe de las piezas usadas en la reparación (sin IVA), fecha de la reparación, hora de inicio de la reparación y hora de finalización de la reparación. Ninguna reparación lleva más de un día. Además de otros métodos que consideres adecuados, la clase debe proporcionar un servicio **total()** que devuelva el total de la reparación (piezas más mano de obra), teniendo en cuenta que se cobran 0,375 € por minuto de mano de obra y que el IVA es del 16%. También habrá un método **mostrar()** que muestre la información de la forma siguiente:

```
jueves 14 de noviembre de 2002
Vehículo: SEAT 600
Propietario: Luis Sanz Sanz
Hora de inicio: 17:15:25
Hora de final: 19:07:12
Importe de las piezas: 213.35 eur
Total de la reparación: 295.771 eur (IVA inc.)
```

Prueba la clase en una función **main()**.

Programación orientada a objetos

Unidad 5 – Taller – Página 2

FdI
UCM

Ejercicio nº 2

Crea una clase **Mueble** que modele la información que se mantiene en una carpintería sobre cada mueble que fabrican: descripción, carpintero (una persona), cantidad de madera utilizada (en m³), día de inicio de la fabricación y día de finalización de la fabricación. Los muebles se tardan en fabricar días completos. Además de otros métodos que consideres adecuados, la clase debe proporcionar un servicio **total()** que devuelva el total de la fabricación (material más mano de obra), teniendo en cuenta que la madera cuesta 1,233 € por m³, que los carpinteros cobran 150 € por día de trabajo y que el IVA es del 16%. También se dispondrá de un método **mostrar()** que muestre la información de la forma siguiente:

```
Mueble: Estantería castellana de tres cuerpos
Carpintero: Luis Sanz Sanz
Día de inicio: 17/11/2002
Día de final: 19/11/2002
Importe de la madera: 1491.93 eur
Importe de la mano de obra: 300 eur
Total de la fabricación: 2078.64 eur (IVA inc.)
```

Prueba la clase en una función **main()**.

Programación orientada a objetos

Unidad 5 – Taller – Página 3

Crea una clase **Pedido** que modele la información que se mantiene en un comercio sobre cada pedido que atienden: código, descripción, cliente (una persona), fecha del pedido, hora del pedido y prioridad (un número). Además de otros métodos que consideres adecuados, la clase debe proporcionar operadores de incremento y decremento que aumenten o decrementen, respectivamente, en una unidad la prioridad del pedido. Y un método **mostrar()** que muestre la información de la siguiente forma:

```
Código: G629X002   Prioridad: 2
Descripción: Fungibles
Cliente: Luis Sanz Sanz (NIF: 223344G)
14/11/2002 17:15:25
```

Prueba la clase en una función **main()**.

Crea una clase **Deuda** que modele la información que mantiene un prestamista sobre cada deuda que contraen con él: deudor (una persona), fecha del último pago, interés diario y deuda (cantidad que se debe). Además de otros métodos que consideres adecuados, la clase debe proporcionar operadores de incremento y decremento que aumenten o decrementen una milésima el interés diario. Además, habrá un método **pago()** que reciba una cantidad a descontar de la deuda y la fecha del pago; antes de descontar la cantidad de lo debido, calculará y sumará los intereses devengados (deuda por interés diario por número de días transcurridos). Por último, habrá un método **mostrar()** que muestre la información de la siguiente forma:

```
Deudor: Luis Sanz Sanz (223344G), 30 años
Deuda pendiente: 3000 eur (Interés: 0.003%)
Desde el lunes 14 de octubre de 2002
```

Prueba la clase en una función **main()**.

Crea una clase **Cuenta** que modele la información que se mantiene en un banco sobre cada cuenta corriente: cliente (una persona), fecha y hora de apertura de la cuenta, y saldo de la misma. Además de otros métodos que consideres adecuados, la clase debe proporcionar operadores suma y resta que devuelvan nuevas cuentas con la cantidad que se pase sumada o restada al saldo de la original. También se quiere disponer de las abreviaturas **+=** y **-=**, así como de un método **mostrar()** que muestre la información así:

```
Cliente: Luis Sanz Sanz (223344G), 30 años
Cuenta abierta el jueves 14 de noviembre de 2002 a las 11:15:25
Saldo actual: 2500 eur
```

Prueba la clase en una función **main()**.

Crea una clase **Evento** que modele la información que se mantiene en un programa sobre los eventos que maneja: responsable (una persona), fecha y hora de inicio del evento, fecha y hora de fin del evento, y nivel de servicio requerido. Además de otros métodos que consideres adecuados, la clase debe proporcionar operadores de incremento y decremento que incrementen o decrementen en una unidad el nivel de servicio. También habrá un método **duracion()** que devolverá la duración del evento en minutos, así como un método **mostrar()** que muestre la información de esta forma:

```
Responsable: Luis Sanz Sanz
Evento iniciado el jueves 13 de junio de 2002 a las 11:15:25
Evento terminado el miércoles 23 de octubre de 2002 a las 19:03:54
Duración: 190548 min.
Nivel de servicio: 8
```

Prueba la clase en una función **main()**.