



FACULTAD DE INFORMÁTICA

Ejercicios de refuerzo

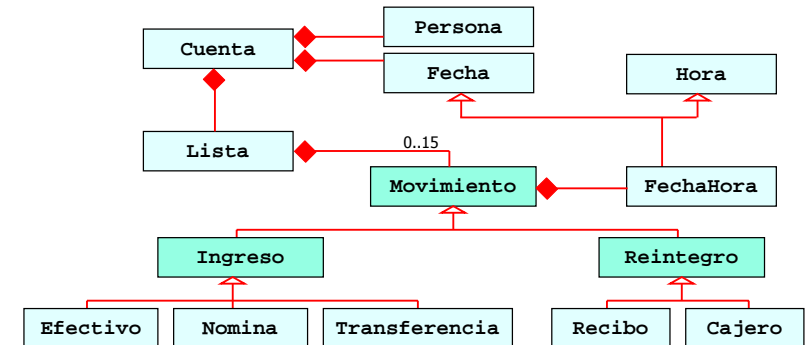
Programación orientada a objetos

Autor: Luis Hernández Yáñez

FdI
UCM

Las clases de partida para los ejercicios

Las clases de partida para los ejercicios son las de la aplicación de cuentas bancarias del último taller:



Excepcionalmente hemos añadido en la clase `Cuenta` un método `bool recuperar(int, Movimiento*&)` y otro `void leer()`.

Programación orientada a objetos

Ejercicios de refuerzo – Página 1

FdI
UCM

Ejercicio nº 1

Desarrolla un programa completo que haga lo siguiente (en el orden que se indica): crear un objeto dinámico `ahorros` de clase `Cuenta`, leer los datos de ese objeto, pedir al usuario los datos de los cuatro primeros movimientos de la cuenta y añadirlos a su lista de movimientos. Finalmente, mostrar en la pantalla toda la información de la cuenta con un formato como el siguiente:

```
Cliente: Alba Santos Torres
NIF: 33445577G
Cuenta abierta el viernes 1 de febrero de 2002
Ultimos movimientos:
  El sabado 2 de febrero de 2002:
    Abono de nomina          32672.15 Eur (11:07:36)
    Cajero automatico        250.00 Eur (13:27:00)
  El martes 5 de febrero de 2002:
    Cajero automatico        300.00 Eur (18:12:16)
  El miercoles 6 de febrero de 2002:
    Transferencia a su favor 1200.25 Eur (09:27:12)
Saldo actual: 241764.46 Eur
```

Se valorará el menor uso de variables puntero (el mínimo es dos).

Programación orientada a objetos

Ejercicios de refuerzo – Página 2

FdI
UCM

Ejercicio nº 2

En la aplicación de cuentas bancarias se hace necesario distinguir entre dos tipos particulares de recibos: de la luz y de una suscripción. Esto no quita que se vayan a seguir usando recibos genéricos.

Para acomodar los tipos concretos de recibos, debes incluir dos clases:

Luz, para los recibos de consumo eléctrico. Además del concepto, la cantidad y el momento necesita dos datos más: el nombre de la compañía eléctrica y los kilowatios-hora consumidos.

Suscripcion, para las suscripciones. Además del concepto, la cantidad y el momento necesita dos datos más: el nombre de la compañía que factura y el número de recibo.

Para estos dos tipos concretos de recibos se mostrará también la información adicional en una segunda línea. En la siguiente página se muestra un ejemplo de salida.

Programación orientada a objetos

Ejercicios de refuerzo – Página 3

Cliente: Alba Santos Torres (33445577G), 25 años
Cuenta abierta el viernes 1 de febrero de 2002
Saldo actual: 207971.42 Eur

02/02/2002 13:27:00	Cajero automatico	250.00 Eur
05/02/2002 18:12:16	Pago de recibo	112.45 Eur
06/02/2002 09:27:12	Pago de recibo de la luz	73.24 Eur
[Hiberdrola, 7.26 Kwh]		
12/02/2002 10:08:23	Pago de suscripcion	34.95 Eur
[Revista PC-World, numero 14]		

Para desarrollar un árbol genealógico, se necesita disponer de una clase **MiembroFamilia** con las siguientes características:

- ✓ Datos sobre cada ejemplar: nombre, apellidos, edad, NIF, fecha de nacimiento, hora de nacimiento, sexo, padre y madre.
- ✓ Los atributos de fecha de nacimiento y hora de nacimiento serán atributos dinámicos.
- ✓ Los atributos de padre y madre serán relaciones con otros objetos de la clase (puede no tenerse información sobre quién es el padre o la madre del miembro de la familia).
- ✓ La clase debe ajustarse a la Forma canónica ortodoxa y disponer de accedentes y mutadores para todos los atributos.
- ✓ La clase debe disponer de un método **leer()** que pida los datos y de un método **mostrar()** que muestre la información de la forma que indica el ejemplo del principio de la siguiente página.

D^a Rosa Gómez Pérez (3456789K)
30 años
Nacida el 12 de abril de 1973 a las 17:07:47
Padre: D. Javier Gómez Ruiz
Madre: D^a Rosa Pérez Galván

- ✓ La clase debe disponer de un método **antepasados()** que muestre todos los antepasados del miembro de la familia. Para cada antepasado se indicará, después del nombre (con el tratamiento oportuno — ver Nota), su fecha y hora de nacimiento:

D. Javier Gómez Ruiz, 25-6-1946 09:23:32
D^a Rosa Pérez Galván, 4-4-1947 22:08:12
...

Se pide desarrollar por completo la clase **MiembroFamilia**.

Nota: siempre que se muestre el nombre de un miembro de la familia se colocará delante el tratamiento que corresponda (D. para los varones y D^a para las mujeres).