



FACULTAD DE INFORMÁTICA

Más sobre clases y objetos

SOLUCIÓN DEL EJERCICIO EXTRA

Programación orientada a objetos — Unidad 2

Autor: Luis Hernández Yáñez

FdI
UCM

Ejercicio

Crea una clase Cuenta (bancaria) con atributos para el número de cuenta (un entero largo), el NIF del cliente (usa la clase de la unidad anterior), el saldo actual y el interés anual que se aplica a la cuenta (porcentaje). Define en la clase los siguientes métodos:

Constructor por defecto y constructor con DNI, saldo e interés.

Accedentes y mutadores.

Para el número de cuenta no habrá mutador.

actualizarSaldo(): actualizará el saldo de la cuenta aplicándole el interés diario (interés anual dividido entre 365 aplicado al saldo).

ingresar(double): permitirá ingresar una cantidad en la cuenta.

retirar(double): permitirá sacar una cantidad (si hay saldo).

Método que nos permita mostrar todos los datos de la cuenta.

El número de cuenta se asignará de forma correlativa a partir de 100001, asignando el siguiente número al último asignado.

Programación orientada a objetos

Unidad 2 – Solución del ejercicio extra – Página 1

FdI
UCM

Cuenta.h

```
#ifndef cuenta_h
#define cuenta_h

#include "nif.h"

class Cuenta {
public:
    Cuenta(long int = 0, double = 0, double = 0);
    Cuenta(const Cuenta&);
    Cuenta& operator=(const Cuenta&);
    ~Cuenta();
    long int numero() const;
    Nif nif() const;
    double saldo() const;
    double interes() const;
    void nif(Nif);
    void saldo(double);
    void interes(double);
```

.../...

Programación orientada a objetos

Unidad 2 – Solución del ejercicio extra – Página 2

FdI
UCM

Cuenta.h

```
void actualizarSaldo();
void ingresar(double);
bool retirar(double);
void mostrar() const;
private:
    static long int _siguiente;
    long int _numero;
    Nif _nif;
    double _saldo;
    double _interes;
};

#endif
```

Programación orientada a objetos

Unidad 2 – Solución del ejercicio extra – Página 3

```
#include <iostream>
using namespace std;

#include "cuenta.h"

long int Cuenta::_siguiente = 100001;

Cuenta::Cuenta(long int dni, double saldo, double interes)
    : _nif(dni), _saldo(saldo), _interes(interres) {
    _numero = _siguiente;
    _siguiente++;
}

Cuenta::Cuenta(const Cuenta& otra) {
    _numero = otra._numero;
    _saldo = otra._saldo;
    _nif = otra._nif;
    _interes = otra._interes;
}

.../...
```

```
Cuenta& Cuenta::operator=(const Cuenta& otra) {
    _numero = otra._numero;
    _saldo = otra._saldo;
    _nif = otra._nif;
    _interes = otra._interes;
    return *this;
}

Cuenta::~Cuenta() { }

long int Cuenta::numero() const { return _numero; }

Nif Cuenta::nif() const { return _nif; }

double Cuenta::saldo() const { return _saldo; }

double Cuenta::interes() const { return _interes; }

void Cuenta::nif(Nif n) { _nif = n; }

void Cuenta::saldo(double s) { _saldo = s; }

void Cuenta::interes(double i) { _interes = i; }

.../...
```

```
void Cuenta::actualizarSaldo() {
    _saldo += _saldo * _interes / 100 / 365;
}

void Cuenta::ingresar(double cantidad) {
    _saldo += cantidad;
}

bool Cuenta::retirar(double cantidad) {
    if(_saldo < cantidad) return false;
    _saldo -= cantidad;
    return true;
}

void Cuenta::mostrar() const {
    cout << "Numero de cuenta: " << _numero << endl;
    cout << "N.I.F.: ";
    _nif.mostrar();
    cout << endl << "Interes anual: " << _interes << "%" << endl;
    cout << "Saldo actual: " << _saldo << " euros" << endl;
}

.../...
```

```
// El programa de prueba - pruebaCuenta.cpp

#include <iostream>
using namespace std;
#include "nif.h"
#include "cuenta.h"

int main()
{
    Cuenta cuenta1(234567, 2000, 7.75);
    cuenta1.mostrar(); cout << endl;
    Cuenta cuenta2(3334555, 1000, 5.25);
    cuenta2.mostrar(); cout << endl;
    Cuenta cuenta3(cuenta1);
    cuenta3.mostrar(); cout << endl;
    Cuenta cuenta4;
    cuenta4.mostrar(); cout << endl;
    cuenta4 = cuenta2;
    cuenta4.mostrar(); cout << endl;

    return 0;
}
```