

2) Programar la clase *Contrato* y sus clases derivadas

La empresa de telefonía desea informatizar los contratos de sus clientes, que puede ser de 2 tipos, contrato de tarifa plana y contrato movil:

- **ContratoTP:** Se paga una cantidad fija al mes, independientemente de la cantidad de llamadas realizadas y el número de minutos hablados hasta un cierto límite. Si el cliente supera el límite de minutos, el exceso de minutos se tarifica a un precio fijo e inamovible de 0.15 céntimos/minuto.

factura= precio tarifa plana + (exceso limite minutos x 0.15)

- **ContratoMovil:** Estos clientes pagan en función del número de minutos que hablan al mes.

factura = precio minuto x minutos hablados

Por cuestiones de estudio de mercado, a la compañía le interesa saber la nacionalidad de este tipo de clientes (cree un atributo de tipo char * en esta clase concreta para guardar la nacionalidad del cliente).

Actualmente la oferta que tiene la compañía para los clientes de tarifa plana es de 300 minutos por 10 euros (programar la clase de forma que cuando la compañía actualice la oferta de tarifa plana los cambios se reflejen en todos los clientes actuales y futuros de la compañía).

En cuanto a los clientes de tarifa móvil el precio por minuto es específico y particular para cada cliente (un cliente puede tener un precio por minuto diferente a otro cliente).

Para cada contrato la empresa desea guardar la siguiente información:

Número contrato (`int idContrato`): debe ser único y generarse automáticamente

dni contrato: (`long int dniContrato`): permite saber a quién pertenece el contrato

Fecha contrato (`Fecha fechaContrato`)

Una vez un contrato es dado de alta en la aplicación, el número de contrato **no puede ser modificado** (debe generarlo automáticamente la aplicación de forma que no puede haber 2 contratos con el mismo número) aunque si puede ser consultado.

El dni asociado al contrato y la fecha del contrato si pueden ser modificados a posteriori (por supuesto, también pueden ser consultados).

Programe una clase **Contrato** para representar un contrato genérico. A partir de dicha clase y mediante herencia programe las clases derivadas **ContratoTP** y **ContratoMovil** de forma que el siguiente código pueda ser ejecutado y produzca la salida siguiente:

Prueba2.cpp

```

#include <cstdlib>
#include <iostream>
#include <iomanip> //std::setprecision
#include "Fecha.h" //definicion de la clase Fecha
#include "Contrato.h" // definicion de la clase Contrato
#include "ContratoTP.h" // definicion de la clase ContratoTP
#include "ContratoMovil.h" // definicion de la clase ContratoMovil

using namespace std;

int main(int argc, char *argv[]) {
    Fecha f1(29,2,2001), f2(f1), f3(29,2,2004);
    cout << "Fechas: ";
    f1.ver(); cout << ", "; f2.ver(); cout << ", "; f3.ver(); cout << endl;

    Contrato *p = new Contrato(75547111, f1), c(23000111, Fecha(2,2,2002));
    cout << ContratoTP::getLimiteMinutos() << " - " << ContratoTP::getPrecio() << endl;
    ContratoTP ct1(17333256, f1, 250); //habla 250 minutos
    ContratoTP ct2(12555100, f3, 320); //habla 320 minutos
    ContratoTP ct3(ct1);
    ContratoMovil cm1(17333256, f1, 0.12, 100, "ESPAÑOL"); //habla 100 minutos
    ContratoMovil cm2(17000000, Fecha(3,3,2003), 0.10, 180, "FRANCES"); //habla 180 minutos
    ContratoMovil cm3(cm2);
    p->ver(); cout << "\n"; c.ver(); cout << endl;
    ct1.ver(); cout << endl; ct2.ver(); cout << "\n"; ct3.ver(); cout << "\n";
    cm1.ver(); cout << endl; cm2.ver(); cout << "\n"; cm3.ver(); cout << "\n";
    cout << p->getIdContrato() << ct2.getIdContrato() << cm2.getIdContrato() << endl;

    cout << setprecision(2) << fixed; //a partir de aqui float se muestra con 2 decimales

    cout << "Facturas: " << ct1.factura() << "-" << ct2.factura() << "-" << cm1.factura() << endl;
    ContratoTP::setTarifaPlana(350, 12); //350 minutos por 12 euros
    p->setDniContrato(cm1.getDniContrato());
    ct3.setFechaContrato(p->getFechaContrato()+1);
    cm3.setNacionalidad(cm1.getNacionalidad());
    cm2.setPrecioMinuto(cm1.getPrecioMinuto()+0.02);
    cm1.setMinutosHablados(ct2.getMinutosHablados()/2);
    ct1.setMinutosHablados(cm3.getMinutosHablados()*2);
    cout << *p << "\n" << c << endl;
    cout << ct1 << endl << ct2 << "\n" << ct3 << "\n" << cm1 << "\n" << cm2 << endl << cm3 << endl;

    system("PAUSE");
    return 0;
}

```

Salida:

```

Fechas: 28/02/2001, 28/02/2001, 29/02/2004
300 - 10

75547111 (1 - 28/02/2001)
23000111 (2 - 02/02/2002)
17333256 (3 - 28/02/2001) 250m, 300(10)
12555100 (4 - 29/02/2004) 320m, 300(10)
17333256 (5 - 28/02/2001) 250m, 300(10)

17333256 (6 - 28/02/2001) 100m, ESPAÑOL 0.12
17000000 (7 - 03/03/2003) 180m, FRANCES 0.1
17000000 (8 - 03/03/2003) 180m, FRANCES 0.1

14
Facturas : 10.00-13.00-12.00

17333256 (1 - 28 feb 2001)
23000111 (2 - 02 feb 2002)
17333256 (3 - 28 feb 2001) 360m, 350(12.00) - 13.50€
12555100 (4 - 29 feb 2004) 320m, 350(12.00) - 12.00€
17333256 (5 - 01 mar 2001) 250m, 350(12.00) - 12.00€

17333256 (6 - 28 feb 2001) 160m, ESPAÑOL 0.12 - 19.20€
17000000 (7 - 03 mar 2003) 180m, FRANCES 0.14 - 25.20€
17000000 (8 - 03 mar 2003) 180m, ESPAÑOL 0.10 - 18.00€

Presione una tecla para continuar . . .

```