

Instrucciones

- Lea con detenimiento y desarrolle **individualmente** cada una de las actividades a realizar durante la experiencia.
- Cree dos archivos. Uno con extensión **.cpp** y otro con extensión **.h** según corresponda con lo desarrollado. El nombre de cada archivo debe tener el siguiente formato:
TEL102_C4_Nombre_Apellido.cpp y **TEL102_C4_Nombre_Apellido.h**, sin tildes. (Ej. TEL102_C4_Patricio_Olivares.cpp y TEL102_C4_Patricio_Olivares.h)
- Enviar el archivo a través de la página de aula del ramo, sección “Control 4” hasta las 11:45:00 del día de **hoy**, Martes 17/11/2020, Hora continental de Chile (UTC-3).
- Cada minuto de atraso tendrá un descuento siguiendo la serie de Fibonacci.
- Trate de utilizar herramientas conocidas o aprendidas en clases. **No copie literalmente de recursos online.**
- Comente adecuadamente el programa, describiendo lo que hace.
- Sea riguroso con las instrucciones de desarrollo.
- ¡Éxito!

Tel-Bank v2.0

Debido a la masiva llegada de nuevos clientes, el banco **Tel-Bank** ha decidido extender sus servicios acompañado de un proceso de mejora de su sistema informático. Impresionado por sus habilidades, nuevamente se han solicitado sus servicios como programador.

Se le entrega una porción de código **control4.h**, (disponible en la página de Aula), el cual será la base para su programa. Este código contiene las clases

- **Cuenta**, la cual contiene información sobre cada cuenta bancaria.
- **Banco**, la cual contiene un vector con todas las cuentas asociadas al banco y las transacciones entre ellas.

grupos.cpp

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

const int cobro = 1000; // Cobro por transaccion realizada
const int saldoMax_cvista = 500000; // Saldo maximo Cuenta Vista
const int retiroMax_cvista = 50000; // Retiro maximo por transaccion Cuenta Vista

class Cuenta{
public:
    Cuenta(int id, int s, int tipo);
    int getId();
    int getSaldo();
    int getTipo();
    bool virtual depositar(int d) = 0; // Deposita cantidad d de dinero
    bool virtual retirar(int d) = 0; // Retira cantidad d de dinero
protected:
    int idCuenta;
    int saldo;
    int tipoCuenta;
};

class Banco{
public:
    Banco(vector<Cuenta*> &c);
    bool transaccion(int id1, int id2,int d);
    void estadoBanco();
private:
    vector<Cuenta*> cuentas;
    int ganancias;
};
```

Tel-Bank desea ofrecer dos tipos de cuentas a sus clientes: **Cuenta Corriente** y **Cuenta Vista**, las cuales cuentan con las siguientes características:

Cuenta Vista

- Tipo de cuenta 0.
- El saldo no puede superar los \$500000 (definidos en la variable `saldoMax_cvista`).
- No puede endeudarse, por lo que su saldo **no puede ser negativo**.
- Se puede retirar un máximo de \$50000 por transacción (definidos en la variable `retiroMax_cvista`).

Cuenta Corriente

- Tipo de cuenta 1.
 - No tiene un límite de saldo.
 - Puede endeudarse, por lo que su saldo **puede ser negativo**.
1. (15pts) En el archivo con extensión `.h`, defina las clases `CuentaCorriente` y `CuentaVista`, las cuales heredan de la clase `Cuenta`.

2. (30pts) En el archivo con extensión `.cpp` implemente los constructores y métodos de las clases, `Cuenta`, `CuentaCorriente` y `CuentaVista` según corresponda. Considere adicionalmente:
 - El constructor de `Cuenta` posee tres parámetros `id`, `s` y `tipo`, que sirven para inicializar los atributos de la clase `idCuenta`, `saldo` y `tipoCuenta` respectivamente.
 - Los métodos `depositar` y `retirar` de las clases `Cuenta` y sus derivadas sirven para incrementar y reducir el saldo de una cuenta. Estas retornan `true` en caso de poder realizarse el depósito o retiro respectivamente. En caso contrario, retorna `false`.
3. (30pts) Implemente el constructor y los métodos de la clase `Banco`. Considere lo siguiente:
 - Existe un único constructor, con un parámetros `c`, el cual es un vector de cuentas utilizado para inicializar el atributo `cuentas`. El banco inicia sin ganancias.
 - El método `transaccion` realiza una transferencia de saldo `d` desde una cuenta de origen `id1` a una cuenta de destino `id2`. Si la transacción pudo efectuarse, se retorna `true`, en caso contrario `false`.
 - Por cada transacción realizada, el banco gana el monto indicado en la variable `cobro`. Este monto se descuenta de la cuenta de origen `id1`.
 - El método `estadoBanco` debe mostrar por pantalla cada una de las distintas cuentas por su `idCuenta`, `tipoCuenta` y su respectivo `saldo`.
4. (25pts) Implemente la función `main`. Esta función debe
 - Crear 4 cuentas con los siguientes datos:
 - Cuenta Vista, `id` = 1124, `saldo` = 400000
 - Cuenta Vista, `id` = 5748, `saldo` = 400000
 - Cuenta Corriente, `id` = 9935, `saldo` = 700000
 - Cuenta Corriente, `id` = 7548, `saldo` = 1500000
 - Cree un banco con las cuentas anteriormente creadas y sin ganancias.
 - Muestre por pantalla el estado inicial de todas las cuentas y las ganancias iniciales
 - Realice (en orden) las siguientes transacciones
 - Transacción 1: `id Origen` = 9935, `id Destino` = 5748, `monto` = 200000
 - Transacción 2: `id Origen` = 9935, `id Destino` = 1124, `monto` = 50000
 - Transacción 3: `id Origen` = 9935, `id Destino` = 7548, `monto` = 800000
 - Transacción 4: `id Origen` = 5748, `id Destino` = 1124, `monto` = 500000
 - Transacción 5: `id Origen` = 9935, `id Destino` = 1124, `monto` = 100000

e indique si cada una de ellas se puede realizar

 - Muestre por pantalla el estado final de todas las cuentas y sus ganancias.

Salida (consola)

```
[elprofe@tel102 control4]$ ./TEL102_C4_Nombre_Apellido
Cuentas creadas:

Cuenta id:1124
Tipo de cuenta:0
Saldo: 400000

Cuenta id:5748
Tipo de cuenta:0
Saldo: 400000

Cuenta id:9935
Tipo de cuenta:1
Saldo: 700000

Cuenta id:7548
Tipo de cuenta:1
Saldo: 1500000

Ganancias: 0

Transaccion 1 no es posible de realizar
Transaccion 2 realizada correctamente
Transaccion 3 realizada correctamente
Transaccion 4 no es posible de realizar
Transaccion 5 no es posible de realizar
Transacciones finalizadas

Cuenta id:1124
Tipo de cuenta:0
Saldo: 450000

Cuenta id:5748
Tipo de cuenta:0
Saldo: 400000

Cuenta id:9935
Tipo de cuenta:1
Saldo: -152000

Cuenta id:7548
Tipo de cuenta:1
Saldo: 2300000

Ganancias: 2000
```