

Instrucciones

- Lea con detenimiento y desarrolle **individualmente** cada una de las actividades a realizar durante la experiencia.
- Cree un archivo con extensión **.cpp** con lo desarrollado. El nombre del archivo debe tener el siguiente formato: **TEL102_C3_Nombre_Apellido.cpp** (Ej. TEL102_C3_Mauricio_Araya.cpp), sin incluir tildes.
- El control está diseñado para ser resuelto en 60 minutos.
- Enviar el archivo a través de la página de aula del ramo, sección “Control 3” hasta las 19:55:00 del día de **hoy**, Miércoles 10/11/2021, 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!

Prepagos Clavistel

La empresa **Clavistel++** tiene el monopolio de las telecomunicaciones en el remota isla de **Seaplusplus**. La telefonía movil funciona solo con esquema de prepago en la isla, es decir, el cliente carga su celular con minutos y los gasta cuando llama a otros teléfonos. Recibir llamadas siempre es posible, independiente del saldo que tenga el teléfono. Por simplicidad, un telefono puede quedar con saldo negativo si es que ocupa más de los minutos disponibles durante la llamada en curso, pero no podrá realizar nuevas llamadas luego de cortar. Los ingenieros de Clavistel++ han diseñado parte del sistema de cobranza y le contratan a usted para continuar con este desarrollo. En particular, se le encomienda implementar un archivo **.cpp** con los métodos de las clases **Telefono** y **Llamada**, las cuales están definidas en el archivo **control3.h**.

control3.h

```
#include<iostream>
#define TELEFONOS 5
using namespace std;

class Telefono{
public:
    Telefono(int prepago): prepago(prepago), ocupado(false),discando(false),
        llamadas(0), entrante(0), saliente(0){}
    int getSaldo();
    void reset();
    bool discar();
    bool recibir();
    void finalizar(int tiempo);
    float getPromedio();
private:
    bool ocupado;
    bool discando;
    int prepago;
    int llamadas;
    int entrante;
    int saliente;
};

class Llamada{
public:
    Llamada(int codigo, Telefono *origen, Telefono *destino);
    void iniciar();
    void finalizar(int tiempo);
private:
    Telefono *origen;
    Telefono *destino;
    int codigo;
};
```

1. (50pts) Implemente los métodos de la clase **Telefono**. Considere lo siguiente:

- Existe un único constructor de la clase que ya está implementado en el .h
- El atributo **prepago** indica los minutos cargados en el teléfono, mientras que **entrante** y **saliente** son la suma de minutos de llamadas recibidas y efectuadas respectivamente. Además, se lleva un registro de la cantidad total de **llamadas** efectuadas y recibidas por el teléfono.
- Un teléfono puede estar solo en una llamada a la vez, por lo que los atributos booleanos¹ **ocupado** y **discando** indican si el teléfono está ocupado por una llamada en este momento y si esta llamada fue discada desde este teléfono respectivamente.
- El método **getSaldo**, obtiene la resta entre los minutos de **prepago** y **saliente**.
- El método **getPromedio** obtiene el promedio de duración en minutos de todas las llamadas (recibidas y efectuadas).
- Los métodos **discar** y **recibir** comienzan una nueva llamada saliente o entrante respectivamente. El teléfono que efectúa la llamada debe verificar si existe saldo para llamar, además de verificar que el teléfono no esté ocupado. En caso contrario, retornará **false**. Quien recibe la llamada, también debe verificar si su teléfono está ocupado. En el caso de que una llamada pueda ser discada pero no recibida por la otra parte, se debe llamar al método **reset** para liberar el teléfono que efectúa la llamada.

¹i.e., binarios: true o false

- Por último, el método **finalizar** registra cuanto **tiempo** en minutos duró la llamada y libera el teléfono de la actual llamada.
2. (25pts) Implemente el constructor y los métodos de la clase **Llamada**. Considere lo siguiente:
- El constructor recibe un número de teléfono (correlativo), y referencias a los objetos **Telefono** de origen y destino.
 - El método **iniciar** establece una nueva conexión entre los teléfonos, y **finalizar** pone término a esta conexión.
3. (25pts) Implemente la función **void printStatus(Telefono *list[])**. Esta función recibe una lista de teléfonos, y muestra por pantalla el número de teléfono (índice), el saldo actual y el promedio de duración de las llamadas para cada teléfono.

Para probar su código, puede ocupar la siguiente función **main**:

control3.cpp

```
int main(){
    Telefono *list[TELEFONOS];
    list[0]=new Telefono(100);
    list[1]=new Telefono(50);
    list[2]=new Telefono(35);
    list[3]=new Telefono(67);
    list[4]=new Telefono(32);
    Llamada call_1(1,list[0],list[1]);
    Llamada call_2(2,list[1],list[2]);
    Llamada call_3(3,list[2],list[0]);
    Llamada call_4(4,list[4],list[3]);
    Llamada call_5(5,list[4],list[0]);
    Llamada call_6(6,list[0],list[2]);
    Llamada call_7(7,list[3],list[0]);
    Llamada call_8(8,list[2],list[3]);
    Llamada call_9(9,list[1],list[2]);
    call_1.iniciar();
    call_2.iniciar();
    call_1.finalizar(50);
    call_2.iniciar();
    call_2.finalizar(40);
    call_3.iniciar();
    call_3.finalizar(40);
    call_4.iniciar();
    call_5.iniciar();
    call_4.finalizar(30);
    call_5.iniciar();
    call_5.finalizar(20);
    call_6.iniciar();
    call_7.iniciar();
    call_6.finalizar(20);
    call_8.iniciar();
    call_9.iniciar();
    call_9.finalizar(10);
    printStatus(list);
    for(int i=0; i<TELEFONOS; i++)
        delete list[i];
    return 0;
}
```

Salida (consola)

```
[elprofe@tel102 control3]$ ./TEL102_Nombre_Apellido
Telefono 0 saldo = 30 promedio = 32.5
Telefono 1 saldo = 0 promedio = 33.3333
Telefono 2 saldo = -5 promedio = 27.5
```

Telefono 3 saldo = 67 promedio = 30
Telefono 4 saldo = -18 promedio = 25