

INFORMÁTICA APLICADA

curso 2006-2007

PRÁCTICA 5: Introducción a C++

El objetivo de esta práctica es introducir al alumno en la programación orientada a objetos en C++. Para ello deberá implementarse una clase en C++ para manejar números complejos.

La práctica consiste en crear la clase **CComplejo** para tratamiento de número complejos y realizar un programa que pruebe el funcionamiento de dicha clase.

Tareas:

1. Un número complejo tiene una parte imaginaria y otra real. Consideraremos la representación cartesiana del número complejo. Por tanto la clase **CComplejo** tendrá dos datos miembro privados para almacenar un número complejo:

```
private:  
    float real; // parte real  
    float imag; // parte imaginaria
```

2. El constructor por defecto de la clase creará el número complejo $0 + 0i$.

```
CComplejo::CComplejo(void)  
{  
    real = 0.0;  
    imag = 0.0;  
}
```

3. Deberá existir un constructor para crear un número complejo a partir de dos números reales (parte real y parte imaginaria), así como a partir de sólo un número real (parte real).

```
CComplejo::CComplejo(float num_real, float num_imag)  
{  
    ...  
}
```

4. Será necesario redefinir el operador asignación para números complejos.

```
CComplejo& CComplejo::operator = (const CComplejo & compl_orig)  
{  
    ...  
}
```

5. Deberán implementarse los siguientes métodos:

- **float Real(void) const;**
Devuelve la parte real del número complejo.
 - **float Imag(void) const;**
Devuelve la parte imaginaria del número complejo.
 - **void AsignarReal(float num_real);**
Asigna un número real a la parte real del número complejo.
 - **void AsignarImag(float num_imag);**
Asigna un número real a la parte imaginaria del número complejo.
 - **void Visualizar(void);**
Muestra en pantalla el número complejo. Ejemplos: **3+3i**, **-5-7i**.
6. Se deben definir las siguientes funciones para realizar operaciones entre números complejos y números reales:
- **CComplejo Sumar(CComplejo a, CComplejo b);**
Suma los números complejos a y b , obteniendo otro número complejo que se devuelve como resultado.
 - **CComplejo Multiplicar(CComplejo a, CComplejo b);**
Multiplica los números complejos a y b , obteniendo otro número complejo que se devuelve como resultado.
 - **CComplejo Multiplicar(float a, CComplejo b);**
Multiplica el número real a por el número complejo b , obteniendo otro número complejo que se devuelve como resultado.
 - **CComplejo Multiplicar(CComplejo a, float b);**
Multiplica el número complejo a por el número real b , obteniendo otro número complejo que se devuelve como resultado.
 - **CComplejo Multiplicar(float a, float b);**
Multiplica los números reales a y b , obteniendo otro número real. Este número real se devolverá transformado a un número complejo, cuya parte real es el producto ab , y cuya parte imaginaria es 0.
7. Escribir la función **main** del programa para mostrar el uso de la clase **CComplejo**.

8. El programa debe utilizar las clases **cin/cout** para el manejo de entrada/salida por consola.
9. Finalmente, se deberá escribir una función main que permita comprobar el funcionamiento del código que se ha escrito. Un ejemplo de esta función se da a continuación:

```
int main(void)
{
    CComplejo c1,c4;           // Constructor por defecto
    CComplejo c2(3,2);        // Constructor con parámetros
    CComplejo c5(3);          // Constructor con parámetros
    CComplejo c3(c1);         // Constructor de copia

    c1.AsignarReal(-5);
    c1.AsignarImag(10);

    // Muestra c5
    c5.Visualizar();

    // Producto de complejos
    c3 = Multiplicar(c1,c2);
    c3.Visualizar();

    // Producto de dos reales
    c4 = Multiplicar(-3,5);
    c4.Visualizar();

    system("pause");

    return 0;
}
```

10. El programa debe seguir el estándar de normalización de la asignatura (directorios, comentarios, configuración, etc.).