

1. Ejercicio 5 del tema 1

Implementa un TAD Complejo, que permita representar números complejos con partes real e imaginaria de tipo `float` y que incluya un constructor parametrizado (2 parámetros: parte real y parte imaginaria), una operación para obtener la parte real y otra para obtener la parte imaginaria, y operaciones de suma (dos versiones: sobrecarga del operador `+` y método `suma`), resta (sobrecarga del operador `-`), multiplicación (sobrecarga del operador `*`) y división (sobrecarga del operador `/`).

Recuerda cómo se opera con n° complejos:

$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

$$(a + bi) - (c + di) = (a - c) + (b - d)i$$

$$(a + bi) \cdot (c + di) = (ac - bd) + (ad + bc)i$$

$$\frac{a+bi}{c+di} = \frac{(a+bi) \cdot (c-di)}{(c+di) \cdot (c-di)} = \frac{(ac+bd) + (bc-ad)i}{c^2+d^2} = \frac{ac+bd}{c^2+d^2} + \frac{bc-ad}{c^2+d^2}i$$

Ten en cuenta que las operaciones de suma, resta, multiplicación y división no modificarán el receptor, crearán y devolverán un resultado de tipo `Complejo`. Y que la operación de división es parcial (no está definida si el divisor es el complejo nulo).

- Completa la declaración de la clase `Complejo` en el archivo `Complejo.h` proporcionado.
- Implementa las operaciones del TAD en el archivo `Complejo.cpp`.
- Desarrolla una función `main` ad-hoc para probar tu TAD: declara e inicializa en ella varios objetos de la clase `Complejo`; prueba las operaciones pedidas, visualizando los resultados mediante la sobrecarga de `<<` proporcionada en `Complejo.h`; y captura y trata la excepción generada por un intento de dividir por el complejo nulo.

2. Ejercicio 6 del tema 1

Partiendo de tu solución al ejercicio anterior, implementa una versión genérica del TAD `Complejo` parametrizable en el tipo de las partes real e imaginaria. Úsala después para trabajar con complejos que usen precisión sencilla (`float`).

3. Extensión del ejercicio 6 del tema 1

Actualiza tu solución del ejercicio anterior a una implementación dinámica del TAD.