Реферативна частина

I.Стандартна бібліотека <complex> на С++

Шаблон класу описує об'єкт, в якому зберігаються два об'єкти типу Type , один з яких представляє дійсну частину комплексного числа, а інший - уявну частину.

Синтаксиc: template <class Type>

class complex

Конструктори:

1. complex - Будує комплексне число із зазначеними дійсною та уявною частинами або як копію деякого іншого комплексного числа.

Визначення типів:

1. value\_type - Тип, що представляє тип даних, який використовується для представлення дійсної та уявної частин комплексного числа.

Функції:

1. imag – повертає уявну частину комплексного числа.
2. real - повертає дійсну частину комплексного числа.

Оператори:

1. operator\* - Перемножує комплексне число на множник, який може бути комплексним або мати той самий тип, що і дійсна та уявна частини комплексного числа.
2. Operator+ - Додає число до комплексного числа, де число, може бути комплексним або того ж типу, що і дійсна і уявна частини комплексного числа, до якого виконується додавання.
3. Operator- - Віднімає число до комплексного числа, де число, може бути комплексним або того ж типу, що і дійсна і уявна частини комплексного числа, до якого виконується додавання.
4. Operator/ - Ділить комплексне число на дільник, який може бути комплексним або мати той же тип, що і дійсна та уявна частини комплексного числа.
5. Operator= - Присвоює число комплексному числу, де число, що присвоюється, може бути комплексним або того ж типу, що і дійсна і уявна частини комплексного числа, якому виконується присвоєння.

Параметри:

1. \_RealVal - Значення дійсноЇ частини, що використовується для ініціалізації комплексного числа, що конструюється.
2. \_ImagVal - Значення уявної частини, що використовується для ініціалізації комплексного числа, що конструюється.
3. complexNum - Комплексне число, дійсна та уявна частини якого використовуються для ініціалізації конструйованого комплексного числа.

II.Стандартна бібліотека <complex.h> на С

Конструктори:

1. complex - Будує комплексне число із зазначеними дійсну та уявною частинами або як копію деякого іншого комплексного числа.

Оператори:

1. operator\* - Перемножує комплексне число на множник, який може бути комплексним або мати той самий тип, що і дійсна та уявна частини комплексного числа.
2. Operator+ - Додає число до комплексного числа, де число, може бути комплексним або того ж типу, що і дійсна і уявна частини комплексного числа, до якого виконується додавання.
3. Operator- - Віднімає число до комплексного числа, де число, може бути комплексним або того ж типу, що і дійсна і уявна частини комплексного числа, до якого виконується додавання.
4. Operator/ - Ділить комплексне число на дільник, який може бути комплексним або мати той же тип, що і дійсна та уявна частини комплексного числа.
5. Operator= - Присвоює число комплексному числу, де число, що присвоюється, може бути комплексним або того ж типу, що і дійсна і уявна частини комплексного числа, якому виконується присвоєння.

Функції:

1. сimag – повертає уявну частину комплексного числа.
2. сreal - повертає дійсну частину комплексного числа.
3. ccos - повертає косинус з комплексного числа.
4. cpow(x,y) – повертає х в степені у де х та y комплексне або дійсне число
5. csqrt – повертає квадратний корінь з комплексного числа.
6. imag – повертає уявну частину комплексного числа.
7. real - повертає дійсну частину комплексного числа.

III.Стандартна бібліотека <tgmath.h> на С

<tgmath.h> заголовок який , крім включення <math.h> і <complex.h> , надає макроси, які викликають відповідну математичну функцію з урахуванням типів параметрів.

Конструктори:

1. complex - Будує комплексне число із зазначеними дійсною та уявною частинами або як копію деякого іншого комплексного числа.

Оператори:

1. operator\* - Перемножує комплексне число на множник, який може бути комплексним або мати той самий тип, що і дійсна та уявна частини комплексного числа.
2. Operator+ - Додає число до комплексного числа, де число, може бути комплексним або того ж типу, що і дійсна і уявна частини комплексного числа, до якого виконується додавання.
3. Operator- - Віднімає число до комплексного числа, де число, може бути комплексним або того ж типу, що і дійсна і уявна частини комплексного числа, до якого виконується додавання.
4. Operator/ - Ділить комплексне число на дільник, який може бути комплексним або мати той же тип, що і дійсна та уявна частини комплексного числа.
5. Operator= - Присвоює число комплексному числу, де число, що присвоюється, може бути комплексним або того ж типу, що і дійсна і уявна частини комплексного числа, якому виконується присвоєння.

Функції:

1. imag – повертає уявну частину комплексного числа.
2. real - повертає дійсну частину комплексного числа.
3. сimag – повертає уявну частину комплексного числа.
4. сreal - повертає дійсну частину комплексного числа.
5. ccos - повертає косинус з комплексного числа.
6. cpow(x,y) – повертає х в степені у де х та y комплексне або дійсне число
7. csqrt – повертає квадратний корінь з комплексного числа.

Опис мого проекта

Проект складається з семи файлів. I\_complex\_h.c та I\_tgmath\_h.c Використовують на Сі заголовочний файл <complex.h> та <tgmath.h> відповідно для розв‘язання, квадатного, кубічного та рівняння четвертого порядку з комплексними коефіцієнтами, а також заміряє швидкодію. I\_comlex.cpp та I\_comlex.h це файл та заголовочний файл використовує на Сі++ стандартну бібліотеку <complex> для вирішення тої самої задачі. Complex.cpp та Comlex.h це файли де описується мій власний клас комлексних чисел. Стандартні операції були перевантажені, також додав esqrt та esbrt – квадратний та кубічний корінь з комлексного числа а також функції введення/виведення та можливість запису в файл, читання з файлу. В файлі Complex\_test.cpp ми порівнюємо результати мого класу з результатами стандартної бібліотеки <complex>, а також тестуємо

функції та методи мого класу.

Я використовував формули для дискримінанта, формули Кардано та формули Ферарі для розв'язання, квадатного, кубічного та рівняння четвертого порядку з комплексними коефіцієнтами.

Самі формули брав звідси: <https://ru.wikipedia.org/wiki/Формула_Кардано>

<https://ru.wikipedia.org/wiki/Метод_Феррари>

<https://ru.wikipedia.org/wiki/Дискриминант>

В ході тестування результати мого класу стандартних бібліотек та заголовчних файлів були однакові. Стосовно швидкодії для цих задач для, як і очікувалося найповільнішим став мій власний клас (від 211 – 286 мл.с.) потім йде стандартна бібліотека <complex> (від 150 – 197 мл.с.) і найшвидшими виявилися заголовочні файли <complex.h> та <tgmath.h> (від 110 – 177 мл.с.) та (від 108 – 150 мл.с.) відповідно.