Київський національний університет імені Тараса Шевченка

Навчально-науковий інститут високих технологій

**РЕФЕРАТ**

з дисципліни «Програмування на С/С++»

на тему:

**Опис бібліотеки complex та порівняння із власним класом**

**Виконав:**

студент 1 курсу

електроніки

Пушкаренко Максим Юрійович

**Перевірив:**

Бородін В.

Київ − 2021

ЗМІСТ

[ВСТУП 3](#_Toc90072725)

[Розділ 1 Порівняння бібліотеки із власним класом 4](#_Toc90072726)

[1.1. Бібліотека 4](#_Toc90072727)

[1.2. Швидкодія 5](#_Toc90072728)

# ВСТУП

**Поскільки ми студенти 1-го курсу і мало вивчаємо мову Сі, був дозвіл не виконувати завдання, які стосуються Сі**

# Порівняння бібліотеки із власним класом

## Бібліотека

Стандартна бібліотека на С++ <complex> має визначення класу complex.

Шаблон виглядить наступним чином: template <class T> class complex, де Т – тип, який може бути в 3 варіаціях: Float, double, long double. Цей клас містить 3 конструктори. 1-й створює об'єкт класу із дійсною частиною real та уявною imaginary. 2-й створює копію об’єкту, а 3-й – об’єкт класу на основі об’єкту, створеного 1-й конструктором.

Операції: “ +, -, \*, /, -=, +=, /=, \*=, =, ==, != “ . Операції більше – менше немає, бо ми не можемо порівнювати, яке комплексне число більше, а яке менше.

Оператори перегружаються 3-ма способами: 1) complex + скаляр, скаляр + complex, complex + complex. Це означає, що ми не можемо ділити або множити комплексне число на скаляр(звичайне дійсне число), а інші операції зі скаляром впливають лише на дійсну частину. В задачі ділення та множення робилося вручну, наприклад std::complex <double> z2(z1.real()/2, z1.imag()/2); = ділення комплексного числа на 2. Як згадувалося раніше, в класі визначені 2 функції-члени: real(), imag(). Тобто реальна і уявна частина відповідно.

Клас комплекс включає в себе деякі математичні функції:

Abs() - абсолютне значення числа

Arg() - аргумент(кут)

Conj() - комплексно – спряжене число

Cos() - косинус числа

Cosh() - гіперболічний косинус числа

Exp() - е в степені комплексного числа

Imag() - уявна частина числа

Log() - натуральний логарифм числа

Log10() - десятковий логарифм числа

Norm() – модуль числа (Ро)

Polar(&module, &argument) – число в полярному виді через модуль і кут

Pow() – число в певній степені

Real() – дійсна частина числа

Sin() – синус числа

Sinh() – гіперболічний синус числа

Sqrt() – квадратний корінь числа

Tan() – тангенс числа

Tanh() – гіперболічний тангенс числа

В програмі найчастіше використовувалися функції real(), imag(), pow(), sqrt(). До речі функція pow() виявилася доволі ефективною, бо для формули Кардано потрібні кубічні корені, які вираховуються з високою точністю та заокругленням. Наприклад в своєму власному класі я, на жаль, так і не зумів розв’язати рівняння 3-го, а тому і 4-го степеня через свій клас, бо для кутів потрібні тригонометричні функції. Також реальні і уявні частини мають множитися на синус/косинус певного кута. Це головна проблема, бо виникали дуже малі числа, які повинні бути нулем, тому нізвідки виникали або уявні, або дійсні частини. Цю проблему потрібно було вирішити, але не хватило часу, хоча на форумах знайти відповіді можна. А з бібліотечним класом complex в цілому було зручно працювати. Спочатку був написаний код для розв’язування задач без комплексних коєфіцієнтів, а лише з дійсними, і інформацію було дуже важко найти, а інформації на рахунок комплексних коєфіцієнтів взагалі не було, хоча й розв’язок виявився коротшим.

## Швидкодія

Для досліду швидкості виконання коду використовувалася бібліотека <chrono>.

Програма вираховувала початковий і кінцевий моменти часу виконання програми і віднімала значення (end – start). В ході перевірок очевидно вияснилося, що програма, побудована на бібліотеці <complex>, швидша за програму, написану за допомогою власного класу Complex. Час визначався в секундах, тому подекуди швидкість бібліотечного класу <complex> становила при автоматичному заокругленні 0 секунд. Подекуди навіть рівняння, розв’язані формулою Кардано швидше вирішувалися бібліотечним класом, ніж квадратне власним класом. Але розв’язування Фераррі вже повинна була використовувати і розв’язування Кардано, тому швидкість виконання в 1.5-2 рази була менше за звичайне квадратне рівняння. Для повноти потрібно було б порівняти ще бібліотеку <complex.h> +<tgmath.h> на Сі, але був дозвіл не виконувати частину Сі.