

# kurs języka C++

## liczby zespolone

Instytut Informatyki  
Uniwersytetu Wrocławskiego

Paweł Rzechonek

---

### Prolog

W średniowieczu próby rozwiązywania równania  $x^2 + 1 = 0$  traktowano jako pozbawione sensu. Od czasu do czasu matematycy dostrzegali fakt, że mogliby skrócić rozważania i otrzymać poprawne wyniki, gdyby w trakcie pracy używali symbolu  $\sqrt{-1}$  i gdyby traktowali go jako zwykłą liczbę. Początek historii liczb zespolonych przypada na wiek XVI – wówczas to dwaj włoscy matematycy N. Tartaglia (1500-1557) i G. Cardano (1501-1576) wykazali, że pierwiastki równania stopnia trzeciego wyrażają się ogólnymi wzorami poprzez współczynniki tego równania. Odkrycie to przyczyniło się jednak do powstania nowego problemu. Okazało się bowiem, że podczas obliczeń mogą pojawić się pierwiastki kwadratowe z liczb ujemnych, a w szczególności  $\sqrt{-1}$ . Oczywiście nie jest to żadna liczba rzeczywista. Ponieważ innych liczb poza rzeczywistymi wówczas nie znano, więc naturalnie pojawiło się pytanie, czym jest pierwiastek kwadratowy z liczby ujemnej. Liczby urojone – taką nazwę nadał im Kartezjusz w XVII wieku (dla kontrastu z liczbami rzeczywistymi). Dodając i mnożąc liczby rzeczywiste i urojone matematycy tworzyli nowe liczby postaci  $(a+bi)$ , gdzie  $i = \sqrt{-1}$ , i które dziś nazywamy *liczbami zespolonymi*.

Arytmetyka liczb zespolonych wcale nie doprowadziła do sprzeczności. W 1748 roku szwajcarski matematyk L. Euler wprowadził je do analizy matematycznej w jednym ze swoich najbardziej znanych dzieł „*Introductio in analysin infinitorum*”, nie tylko nie dochodząc do sprzeczności, lecz powodując istotny postęp analizy. W XIX wieku powstał też dział matematyki oparty w całości na liczbach zespolonych – analiza zespolona, która również znalazła zastosowania w matematyce i poza nią. Pierwszą poprawną publikacją na temat całkowania w dziedzinie liczb zespolonych była praca Cauchy'ego z 1825 roku. Jednym z problemów analizy zespolonej jest hipoteza niemieckiego matematyka B. Riemann'a wysunięta w XIX wieku, istotna dla analitycznej teorii liczb. Uznano ją za jedno z najdonioślejszych zagadnień całej matematyki, przez co znalazła się na listach:

- 23 problemów Hilberta w roku 1900;
- siedmiu problemów milenijnych w roku 2000.

Liczby zespolone to przykład pojęcia, które po setkach lat od odkrycia znalazło swoje główne zastosowanie i okazało się być fundamentalne dla techniki, przede wszystkim elektrotechniki.

## Zadanie 1

Zdefiniuj klasę `complex` reprezentującą liczbę zespoloną w postaci algebraicznej ( $a + bi$ ), gdzie parametry  $a$  i  $b$  to liczby rzeczywiste. W klasie `complex` zdefiniuj konstruktor z argumentami domyślnymi tak, aby można go było wywołać bez żadnych argumentów. Poza klasą `complex` zdefiniuj podstawowe operatory arytmetyczne (dodawanie, odejmowanie, mnożenie, dzielenie) oraz operatory strumieniowe do wczytywania i wypisywania liczb zespolonych.

```
class complex {
    double r, i;
public:
    complex(double a = 0, double b = 0) : r(a), i(b) { }
public:
    double re() const;
    double im() const;
    void re(double r);
    void im(double i);
    complex conjugated() const;
};

complex operator+(const complex &x, const complex &y);
complex operator-(const complex &x, const complex &y);
complex operator*(const complex &x, const complex &y);
complex operator/(const complex &x, const complex &y);
std::ostream& operator<<(std::ostream &os, const complex &c);
std::istream& operator>>(std::istream &is, complex &c);
```

Klasę `complex` i związane z nią operatory umieść w przestrzeni nazw `math`.

Na koniec definicję klasy `complex` oraz związanych z nią operatorów arytmetycznych i strumieniowych umieść w bibliotece łączonej dynamicznie – w systemie Windows będzie to plik `.dll` a w systemie Linux `.so`.

## Zadanie 2

Zdefiniuj klasę `polynomial` do przechowywania wielomianu zespolonego o określonym stopniu oraz określonych współczynnikach. Zaprojektuj tą klasę tak, aby stopień wielomianu oraz jego współczynniki były niepubliczne (współczynniki pamiętaj w tablicy liczb zespolonych utworzonej na stercie). Należy zapewnić metody pozwalające odczytywać i ustawiać te pola: do odczytu stopnia wielomianu zdefiniuj składową funkcję dostępową `a` do odczytu i zapisu poszczególnych współczynników zdefiniuj operatory indeksowania.

```
class polynomial {
    int n; // stopień wielomianu
    complex *a; // współczynniki wielomianu;
    // ...
};
```

Współczynnik zespolony  $a_i$  niech będzie pamiętany na pozycji  $i$ -tej w tablicy  $a$  (czyli w komórce  $a[i]$ ) dla  $i = 0 \dots n$ .

W klasie `polynomial` zdefiniuj konstruktor bezargumentowy, konstruktor z listą inicjalizującą (ze współczynnikami zespolonymi), zaimplementuj kopowanie i przenoszenie (dotyczy konstruktorów i operatorów przypisania) oraz destruktor (usunięcie tablicy współczynników wielomianu).

Zdefiniuj operatory dodawania i odejmowania wielomianów, operator mnożenia wielomianu przez stałą zespoloną i mnożenia przez inny wielomian (możesz do kompletu zdefiniować operatory dzielenia wielomianów, jeśli chcesz uzupełnić definicję, ale to nie jest wymagane) oraz operator wywołania funkcji obliczający wartość wielomianu w zadanym punkcie zespolonym za pomocą *schematu Hornera*.

Klasę `polynomial` i związane z nią operatory umieść w przestrzeni nazw `calc`.

Na koniec definicję klasy `polynomial` oraz związanych z nią operatorów arytmetycznych umieść w bibliotece łączonej dynamicznie – w systemie Windows będzie to plik `.dll` a w systemie Linux `.so`.

### **Zadanie 3**

Na koniec napisz program, który rzetelnie przetestuje całą funkcjonalność liczby zespolonej i wielomianu zespolonego. W jednym z testów zrealizuj zadanie wyznaczania miejsc zerowych zespolonej funkcji kwadratowej, której współczynniki poda użytkownik w czasie wykonania.

#### **Uwaga**

Definiując wielomian zespolony `polynomial` w zadaniu drugim wykorzystaj klasę `complex` zdefiniowaną w zadaniu pierwszym.

#### **Ważne elementy programu**

- Tworzenie bibliotek łączonych dynamicznie.
- Korzystanie z bibliotek łączonych dynamicznie.
- Definicje operatorów arytmetycznych i strumieniowych.
- Zgłaszanie wyjątków w sytuacjach błędnych.
- Program testujący w funkcji `main()`.