

C++

liczby zespolone (klasa z operatorami)

studia wieczorowe

Zdefiniuj klasę `LZesp` reprezentującą liczbę zespoloną. Należy zdefiniować dla tej klasy operatory arytmetyczne `+`, `-`, `*`, `/`, operatory porównania `==` i `!=` oraz metodę obliczającą moduł (według podanej definicji).

Liczbą zespoloną nazywamy uporządkowaną parę liczb rzeczywistych. Niech $z = (x, y)$ będzie liczbą zespoloną. Wówczas x nazywamy *częścią rzeczywistą* liczby zespolonej i oznaczmy ją przez $Re(z)$, a y *częścią urojoną* liczby zespolonej i oznaczmy ją przez $Im(z)$. Zbiór wszystkich liczb zespolonych to $\mathbf{C} \stackrel{def}{=} \{(x, y) : x, y \in \mathbf{R}\}$.

Z działaniami dodawania i mnożenia określonymi jak niżej zbiór liczb zespolonych \mathbf{C} tworzy ciało, przy czym rolę zera pełni $(0, 0)$ a rolę jedyki pełni $(1, 0)$.

Liczby zespolone można ze sobą porównywać, wykonywać na nich podstawowe operacje arytmetyczne (dodawanie, odejmowanie, mnożenie, dzielenie) oraz obliczać *moduł* (funkcję tą interpretuje się jako liczbę rzeczywistą mierzącą odległość zadanej liczby zespolonej od punktu $(0, 0)$ na płaszczyźnie zespolonej). Niech $z_1 = (x_1, y_1)$ i $z_2 = (x_2, y_2)$ będą liczbami zespolonymi. Wówczas:

- równość $z_1 = z_2 \stackrel{def}{\iff} x_1 = x_2 \wedge y_1 = y_2$
- moduł liczby zespolonej $z = (x, y) \quad |z| \stackrel{def}{=} \sqrt{x^2 + y^2} \in \mathbf{R}$
- dodawanie $z_1 + z_2 \stackrel{def}{=} (x_1 + x_2, y_1 + y_2)$
- dla każdej liczby zespolonej $z = (x, y)$ istnieje liczba przeciwna $-z \stackrel{def}{=} (-x, -y)$
- odejmowanie (dodanie liczby przeciwnej) $z_1 - z_2 \stackrel{def}{=} z_1 + (-z_2) = (x_1 - x_2, y_1 - y_2)$
- mnożenie $z_1 \cdot z_2 \stackrel{def}{=} (x_1x_2 - y_1y_2, x_1y_2 + y_1x_2)$
- dla każdej liczby zespolonej $z = (x, y) \neq (0, 0)$ istnieje liczba odwrotna $\frac{1}{z} \stackrel{def}{=} \left(\frac{x}{x^2+y^2}, \frac{-y}{x^2+y^2} \right)$
- dzielenie (mnożenie przez liczbę odwrotną) $\frac{z_1}{z_2} \stackrel{def}{=} z_1 \cdot \frac{1}{z_2} = \left(\frac{x_2x_1+y_2y_1}{x_2^2+y_2^2}, \frac{x_2y_1-y_2x_1}{x_2^2+y_2^2} \right)$

Operatory arytmetyczne zdefiniuj w taki sposób, aby można było ich używać także z argumentami typu `double` (konstruktor konwertujący) po obu stronach tego operatora (przyjaźń).

```
LZesp a, b;
// ...
a+b;
a+3;
5+b;
```

Obiekty klasy `LZesp` powinno się dać w łatwy sposób wykorzystać w wyrażeniach warunkowych (nie próbuj definiować operatora konwersji `double()` bo pojawią się niejednoznaczności w wyrażeniach arytmetycznych). Można się na przykład posłużyć operatorem wyłuskania `*` sprawdzającym, czy wartość modułu liczby zespolonej jest różna od 0.

```
LZesp c ;  
// ...  
if (*c) // c!=(0,0)  
{ /*...*/ }  
else // c==(0,0)  
{ /*...*/ }
```

Zaprogramuj także zaprzyjaźnione operatory do czytania ze strumienia wejściowego `operator>` i pisania do strumienia wyjściowego `operator<`.

Całą definicję klasy `LZesp` podziel na część nagłówkową i źródłową. Następnie w osobnym pliku umieść program testowy, który sprawdzi poprawność zdefiniowanych przez Ciebie operacji na liczbach zespolonych.

Uwaga! Klasa `LZesp` będzie wykorzystywana w następnych zadaniach, a zatem warto napisać ją bardzo starannie.