

# kurs języka C++

## liczby wymierne

Instytut Informatyki  
Uniwersytetu Wrocławskiego

Paweł Rzechonek

---

### Prolog

Liczba wymierna to taka liczba, którą można zapisać w postaci ułamka zwykłego, czyli w postaci  $p/q$ , gdzie  $p$  to dowolna liczba całkowita a  $q$  to liczba całkowita różna od 0. Zbiór wszystkich liczb wymiernych oznaczamy symbolem  $\mathbb{Q}$  i formalnie można go zdefiniować jako:

$$\mathbb{Q} = \{p/q : p, q \in \mathbb{Z} \wedge q \neq 0\}$$

Liczby wymierne z operacją dodawania (element neutralny dodawania to zero) i mnożenia (element neutralny mnożenia to jeden) stanowią ciało. Szczególnym przypadkiem liczb wymiernych są liczby całkowite.

### Zadanie

Zdefiniuj klasę `wymierna`, reprezentującą liczbę wymierną w postaci pary liczb całkowitych: licznika i mianownika.

```
class wymierna {
    int licz, mian;
    // ...
};
```

Zadbaj o to, aby mianownik zawsze był liczbą dodatnią oraz aby największy wspólny dzielnik licznika i mianownika zawsze był równy 1. Udostępnij też gettery, czyli funkcje składowe umożliwiające odczyt licznika i mianownika. Definicję liczby wymiernej umieść w przestrzeni nazw obliczenia.

Klasa `wymierna` powinna być wyposażona w konstruktor z licznikiem i mianownikiem, konstruktor konwertujący z wartości typu `int` (możesz zaadoptować do tego celu poprzedni konstruktor definiując drugi argument jako domyślny).

W klasie `wymierna` zdefiniuj operatory binarne umożliwiające wykonywanie obliczeń arytmetycznych (dodawanie `+`, odejmowanie `-`, mnożenie `*` i dzielenie `/`) oraz jednoargumentowe operatory `-` do zmiany znaku na przeciwny i `!` do wyznaczenia odwrotności (zamiana licznika z mianownikiem z pozostawieniem znaku liczby w liczniku). Zdefiniuj także operator rzutowania na typ `double` oraz operator jawnego rzutowania na typ `int` (zaokrąglenie do najbliższej liczby całkowitej).

Nie zapomnij przy każdej funkcji składowej, przy konstruktorach i przy operatorach zadeklarować czy zgłaszają one jakieś wyjątki czy nie. Na przykład, przy dzieleniu przez 0 należy zgłosić wyjątek reprezentujący dzielenie przez 0; w przypadku operacji arytmetycznych dodawania, odejmowania i mnożenia, gdy wynik nie będzie mógł być wyrażony dokładnie jako iloraz dwóch liczb typu `int` należy zgłosić wyjątek reprezentujący przekroczenie zakresu. Zaprojektuj zatem hierarchię klas wyjątków na potrzeby liczb wymiernych zaczynając od klasy bazowej `wyjątek_wymierny` dziedziczącej po `std::logic_error`.

Zaprogramuj także operator strumieniowy do zapisania liczby wymiernej do strumienia wyjściowego operator `<<` w postaci liczby rzeczywistej w systemie dziesiętnym z ułamkiem okresowym.

```
class wymierna {
    // ...
    friend ostream& operator<< (ostream &wyj, const wymierna &w);
};
```

Na koniec napisz program, który rzetelnie przetestuje wszystkie metody z klasy `wymierna` (wraz ze zgłaszanymi przez nie wyjątkami).

### Ważne elementy programu

- Użycie przestrzeni nazw obliczenia.
- Konstruktory i destruktory w klasie `wymierna`.
- Operatory rzutowania i konstruktory konwertujące w liczbie wymiernej.
- Definicja operatorów arytmetycznych dla liczb wymiernych.
- Tekstowa reprezentacja liczby wymiernej w postaci ułamka okresowego.
- Definicja hierarchii klas wyjątków dziedziczących po `std::logic_error`.
- W funkcji `main()` należy przetestować całą funkcjonalność liczby wymiernej.