

# 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.

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) oraz konstruktor domyślny, ustawiający wartość liczby wymiernej na zero (czyli ułamek 0/1).

W klasie `wymierna` zdefiniuj operatory arytmetyczne, które będą wykonywały podstawowe operacje arytmetyczne na liczbach wymiernych: operatory binarne, takie jak dodawanie `+` i `+=`, odejmowanie `-` i `-=`, mnożenie `*` i `*=`, dzielenie `/` i `/=` oraz operatory unarne – do zmiany znaku na przeciwny i `!` do wyznaczenia odwrotności (zamiana licznika z mianownikiem oraz pozostawienie znaku liczby w liczniku); operatory występujące w połączeniu z przypisaniem niech będą operatorami składowymi, a pozostałe to operatory zaprzyjaźnione z klasą.

Zdefiniuj także operator domniemanego 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. W przypadku binarnych operacji arytmetycznych należy zgłosić wyjątek `przekroczenie_zakresu`, gdy wynik nie będzie mógł być wyrażony jako iloraz dwóch liczb typu `int`; w przypadku dzielenia przez 0 podczas operacji dzielenia lub podczas liczenia odwrotności należy zgłosić wyjątek `dzielenie_przez_0`. Zaprojektuj zatem własną hierarchię klas wyjątków na potrzeby liczb wymiernych zaczynając od klasy bazowej `wyjatek_wymierny` dziedziczącej po `std::logic_error`.

Zdefiniuj także zaprzyjaźniony operator strumieniowy `operator<<` do zapisania liczby wymiernej do strumienia wyjściowego w postaci liczby rzeczywistej w systemie dziesiętnym z ułamkiem okresowym. Przykładowo ułamek  $2359348/99900$  należy przedstawić w postaci  $23.61(709)$ .

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

Na koniec napisz program, który rzetelnie przetestuje wszystkie funkcje składowe i operatory z klasy `wymierna` (wraz ze zgłaszanymi przez nie wyjątkami). Przetestuj także kopiowanie liczb wymiernych (domyślny konstruktor kopiujący i przypisanie kopiujące).

### Uwaga

Podziel program na pliki nagłówkowe (definicje klas) i źródłowe (definicje metod zadeklarowanych w klasach). Funkcję `main()` z testami umieść w osobnym pliku źródłowym.

Definicję liczby wymiernej wraz z dedykowanymi wyjątkami umieść w przestrzeni nazw `obliczenia`.

### Ważne elementy programu

- Użycie przestrzeni nazw `obliczenia`.
- Operatory rzutowania i konstruktor konwertujący.
- Definicja operatorów arytmetycznych binarnych i unarnych.
- Zapis do strumienia liczby wymiernej w postaci ułamka okresowego.
- Definicja własnej hierarchii klas wyjątków dziedziczących po `std::logic_error`.
- Zgłaszanie wyjątków w konstruktorach, funkcjach składowych i operatorach.
- Podział programu na pliki nagłówkowe i pliki źródłowe (wyodrębniony osobny plik z funkcją `main()` z testami).