

Programowanie obiektowe C++

Przeciążanie operatorów, konstruktory

Obiekty, konstruktory, operatory

1. Przykład

Dana jest klasa do opisu ułamków zwykłych i podstawowych operacji arytmetycznych na ułamkach:

```
class TFraction
{
public:
    TFraction(int num = 1, int den = 1);
    TFraction(const TFraction &f);

    TFraction operator + ( ) const;
    TFraction operator - ( ) const;
    TFraction & operator = (const TFraction &f);
    TFraction operator + (const TFraction &f) const;
    TFraction operator - (const TFraction &f) const;
    TFraction operator * (const TFraction &f) const;
    TFraction operator / (const TFraction &f) const;

    void reduce();
    string toString() const;
    double value() const;

    void setNum(int num);
    void setDen(int den);

    int getNum() const;
    int getDen() const;

private:
    int num;
    int den;

    void fixZero();
    void fixSign();
};
```

w skład której wchodzi:

- prywatne pola: num i den – do zapamiętywania wartości licznika i mianownika ułamka,
- konstruktor z parametrami domyślnymi **TFraction()** pozwalający na zdefiniowanie domyślnej wartości ułamka w postaci $\frac{1}{1}$, ponadto konstruktor pozwala na zdefiniowanie liczby całkowitej w postaci $\frac{\text{licznik}}{1}$, i dodatkowo umożliwia zdefiniowanie ułamka zwykłego w postaci $\frac{\text{licznik}}{\text{mianownik}}$
- dwie metody **setNum()** i **setDen()** do zmiany wartości licznika i mianownika ułamka,

- metoda **toString()** zwracająca wartość ułamka w łańcuchu znaków do wyświetlania,
- metoda **value()** zwracająca numeryczną wartość ułamka,
- metoda **reduce()** służąca do skracania ułamka (wykorzystuje algorytm szukania największego wspólnego dzielnika dwóch liczb),
- metoda **fixZero()** służąca do zmiany zerowych ułamków na postać $\frac{0}{1}$,
- metoda **fixSign()** przenosząca znak minus z mianownika do licznika,
- operatory jednoargumentowe **+** i **-** do zmiany znaku ułamka,
- dwuargumentowe operatory arytmetyczne: **+**, **-**, ***** i **/**,
- operator wyjścia **<<** pozwalający na wyświetlanie ułamka.

Wygeneruj nowy projekt C++ i skopiuj do pliku źródłowego projektu zawartość pliku tekstowego **ułamki.cpp** ze strony przedmiotu. Uruchom program, prześledź jego działanie.

2. Zadanie

Wykonaj kilka dowolnych operacji arytmetycznych na nowo zdefiniowanych obiektach typu **TFraction**. Wykonaj dowolne operacje arytmetyczne na tablicach obiektów **TFraction**.

3. Zadanie

Dopisz dodatkową funkcję operatorową dla operatora wejścia **>>**, która pozwoli na zmianę wartości ułamka wykorzystując wartości ze standardowego wejścia.

Zadania dodatkowe

1. Opracuj typ obiektowy i program do przechowywania i podstawowych operacji arytmetycznych dla liczb zespolonych w postaci: $\Re + \Im \cdot i$.
2. Opracuj typ obiektowy i program do przechowywania i podstawowych operacji na danych reprezentujących czas. Ogranicz się do godzin i minut. Przykładowa deklaracja typu mogłaby wyglądać:

```
class TTime
{
public:
    // stosowne konstruktory

    void addHour(int h);
    void addMinute(int m);
    void timeReset(...); // uzupełnic
    void timeShow(...) const; // uzupełnic

    TTime operator +(const TTime &t) const;
    TTime operator -(const TTime &t) const;

private:
    int hours;
    int minutes;
};
```

3. Opracuj typ obiektowy i program do przechowywania i podstawowych operacji arytmetycznych dla długich liczb całkowitych (więcej niż 20 cyfr w liczbie).