

Paradygmaty programowania - studia stacjonarne, lista 10, 14.12.2017, godzina 15:15
Czas na rozwiązanie: 60 min.

WAŻNE1: Do każdego zadania przygotuj 3-5 testów sprawdzających poprawność działania!

WAŻNE2: Nie wolno stosować funkcji bibliotecznych i konstrukcji innych niż te przedstawione na wykładach 1-9.

1. Zaimplementować klasę ułamków prostych. Konstruktor klasy powinien sprawdzać czy tworzone ułamki są prawidłowe, na przykład mianownik nie jest liczba zero lub ujemną.

Ważne jest też to aby ułamek był zawsze w najprostszej postaci (to znaczy podzielić licznik i mianownik przez największy wspólny dzielnik - patrz przykładowa implementacja *nwd(x,y)*)

Klasa powinna zapewniać co najmniej realizację operacji dodawania, odejmowania, mnożenia i dzielenia oraz odwracania ułamków prostych. Klasa powinna także sprawdzać czy dwa ułamki są sobie równe i oczywiście nie zapominamy o “ładnej reprezentacji” ułamków na konsoli, to jest przygotowujemy nową wersję *toString()*.

Obiekty klasy nie mogą być modyfikowalne (ang. immutable class), to znaczy że wyniku każdej z tej operacji powstaje nowy obiekt klasy będący jej wynikiem. Co więcej aby “zapobiec” dodania, przez innych programistów, do klasy metod, które zmieniają jej stan, należy zabronić dziedziczenia po klasie ułamków prostych.

Wykorzystując zaimplementowaną klasę napisać program ilustrujący powyższe operacje.

```
int nwd(int x, int y) {
    while (x != y) {
        if (x > y)
            x -= y;
        else
            y -= x;
    }
    return x;
}
```

Punkty: 20 (język Scala)