Programowanie zaawansowane, lab. 4

1. Zdefiniuj klasę reprezentującą liczby wymierne. Musi ona zawierać licznik i mianownik będące liczbami całkowitymi. Utwórz odpowiedni konstruktor przyjmujący dwie liczby całkowite w argumentach. Wszystkie pola powinny być prywatne. Pamiętaj o utworzeniu odpowiednich getterów i seterów, a także o tym, że mianownik nie może wynosić 0. Możesz też dopisać własne pola pomocnicze.
2. Nadpisz dla liczb wymiernych metody:
    - **public boolean equals(Object o)** – metoda ta ma być obsłużona dla innych liczb wymiernych oraz dla podstawowych typów liczbowych występujących w Javie: int, float, double, short, long. Przykładowo 5/1 jest równe liczbie 5.0 (float), a także 5 (int). Pamiętaj, że liczba może być zapisana w postaci 10/2, a też jest równa 5!
    - **public String toString()** – metoda ma wypisywać ułamki właściwe w postaci licznik/mianownik. Części całkowite ułamków niewłaściwych mają być wyciągane przed kreskę ułamkową, np. -3 2/7. Liczby całkowite zapisuj bez kreski ułamkowej, tj. 5 zamiast 5/1. Jeżeli liczba jest ujemna, umieszczaj minus przed częścią całkowitą lub przed częścią ułamkową (-1/2, a nie 1/-2).
3. Zdefiniuj podstawowe operacje na liczbach wymiernych:
    - **public LiczbaWymierna dodaj(LiczbaWymierna w);**
    - **public LiczbaWymierna odejmij(LiczbaWymierna w);**
    - **public LiczbaWymierna pomnoz(LiczbaWymierna w);**
    - **public LiczbaWymierna podziel(LiczbaWymierna w);**
    - **public LiczbaWymierna odwroc() throws ArithmeticException;**
    - **public LiczbaWymierna skroc();**

    Odwrotność ma rzucać **ArithmeticException**, jeśli po odwróceniu mianownik będzie wynosił 0. Do części z powyższych operacji przydatne będzie zaimplementowanie metod liczących NWD oraz NWW. Do policzenia NWD możesz wykorzystać algorytm Euklidesa, ale pamiętaj, że pracuje on tylko na liczbach dodatnich. Zakładamy, że etap konstruowania liczby pozwala na zapisanie jej w dowolnej postaci, niekoniecznie skróconej, np. 2/3, 4/6, 16/24 itd. Jeśli natomiast zostanie dla tej liczby wywołana któraś z czterech pierwszych metod (dodawanie, odejmowanie, mnożenie lub dzielenie), ma ona też zostać automatycznie skrócona.

4. W metodzie **main()** utwórz kilka różnych liczb wymiernych, o dodatnich i ujemnych licznikach i mianownikach. Część z nich niech będzie > 1, część < 1. Wypisz te liczby, przetestuj na nich utworzone w zad. 3 operacje i wypisz wyniki. Przetestuj również działanie metody **equals()** porównując wybraną liczbę z inną liczbą oraz z typami prostymi, tj. int, float itd. Na koniec oblicz wartość dwóch wyrażeń:

$$1)\ \frac{\frac{1}{2}+\frac{-2}{4}}{\frac{3}{4}-\frac{5}{6}} \quad 2)\ \frac{\frac{-3}{2}+\frac{2}{7}}{\frac{3}{5}\cdot\frac{4}{3}}$$

    Jeśli w pierwszym wyrażeniu otrzymałeś wynik 0, a w drugim -1 29/56, jest duża szansa, że liczby wymierne zostały zaimplementowane poprawnie!