

Задача 1

1. Създайте клас **Rational**, представящ рационално число.
 - Създайте конструктор с два параметъра, инициализиращ числителя и знаменателя на рационално число.
 - Напишете методи, които реализират операторите $+$, $-$, $*$, $/$, $==$, $!=$, $<$, $>$.
 - Предефинирайте методите `equals`, `hashCode` и `toString`.
2. Направете класа **Rational** да наследява интерфейса **Comparable**.
 - Реализирайте метода `compareTo` на интерфейса **Comparable**.
3. Създайте собствено изключение **RationalException**, представящо грешки при работа с рационални числа.
4. Направете следните промени в класа **Relational**:
 - Конструкторът с два параметъра хвърля **RationalException**, ако знаменателят на рационалното число е равен на нула или ако и числителят и знаменателят са отрицателни числа (приемете, че числителят носи знака на рационалното число).
 - Напишете конструктор по подразбиране, който въвежда стойности за числителя и знаменателя от клавиатурата; прехвърля обратно в извикващия стек изключение, ако потребителят въведе некоректни целочислени стойности за числителя и знаменателя; хвърля **RationalException**, ако знаменателят на рационалното число е нула или и числителят и знаменателят са отрицателни числа.
5. Създайте драйверен клас, който тества операциите с рационални числа.
 - Добавете статичен метод с три параметъра: ляв операнд, десен операнд и операция, който изпълнява разрешените операции с рационални числа и хвърля **NoSuchMethodException**, ако операцията не е разрешена.
 - Тествайте всички операции с рационални числа.
 - Въведете масив от рационални числа и го сортирайте в нарастващ ред, като използвайте метода `Array.sort`.