

1. Създайте клас **РАЦИОНАЛНО ЧИСЛО**.
 - а) създайте конструктор с два параметъра;
 - б) предефинирайте операторите **+**, **-**, *****, **/**, **==**, **!=**, **<**, **>**;
 - в) предефинирайте методите **Equals**, **GetHashCode** и **ToString**.
2. Направете класа **РАЦИОНАЛНО ЧИСЛО** да наследява интерфейса **IComparable** или **ICompare**.
 - а) реализирайте метода **CompareTo** на интерфейса **IComparable** или метода **Compare** на интерфейса **ICompare**;
3. Създайте собствено изключение при работа с рационални числа.
4. Направете следните промени в класа **РАЦИОНАЛНО ЧИСЛО**:
 - а) конструкторът с два параметри хвърля собственото изключение, ако знаменателят на рационалното число е нула или и числителят и знаменателят са отрицателни числа (приемете, че числителят носи знака на рационалното число);
 - б) напишете конструктор без параметри, който въвежда стойности от клавиатурата за числител и знаменател; прехвърля изключение в стека при неправилно въведени целочислени стойности за числител и знаменател; хвърля собственото изключение, 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.
 - а) добавете статичен метод с три параметъра: ляв операнд, десен операнд и операция, който извършва разрешените операции с рационални числа, като вдига собственото изключение при неразрешена операция;
 - б) тествайте операциите с рационални числа;
 - в) въведете масив от рационални числа и го сортирайте във възходящ ред, като използвате метода **Array.Sort**.