

Упражнение №8

Обработка на изключения

- I. Дефинирайте собствено изключение **RationalNumberException** при работа с рационални числа чрез наследяване на класа **Exception**.
- II. Създайте клас **Rational** (**РАЦИОНАЛНО ЧИСЛО**), който реализира интерфейса **Comparable**.
 1. Декларирайте две целочислени полета с **private** достъп: числител **numerator** и знаменател **denominator** от тип **double**.
 2. Добавете конструктор с два параметъра за инициализиране на полетата.
 3. Добавете метод за достъп **getNumerator**, който връща числителя от тип **double**.
 4. Добавете метод за достъп **getDenominator**, който връща знаменателя от тип **double**.
 5. Напишете метод **add** с два параметъра от тип **Rational**, който реализира операцията **събиране** на рационални числа.
 6. Напишете метод **subtract** с два параметъра от тип **Rational**, който реализира операцията **изваждане** на рационални числа.
 7. Напишете метод **multiply** с два параметъра от тип **Rational**, който реализира операцията **умножение** на рационални числа.
 8. Напишете метод **divide** с два параметъра от тип **Rational**, който реализира операцията **деление** на рационални числа.
 9. предефинирайте метода **toString** на класа **Object**, който дава **СИМВОЛНОТО ПРЕДСТАВЯНЕ НА РАЦИОНАЛНО ЧИСЛО** – числител/знаменател.
 10. Реализирайте метода **compareTo** на интерфейса **Comparable** (**java.lang**)

```
public int compareTo (Object o)
```

който сравнява текущия обект с обекта **o** от класа **Object** и връща отрицателно число, 0 или положително число, ако текущият обект е по-малък, равен или по-голям от параметъра **o**, съответно.
- III. Проектирайте и реализирайте аplet, който тества операциите с рационални числа, като реализира интерфейса **ActionListener**.
 1. Декларирайте полетата и компонентите на графичния потребителски интерфейс с **private** достъп:
 - а) масив от рационални числа;
 - б) две текстови полета за въвеждане на числител и знаменател със съответните етикети пред тях;
 - в) шест бутона, при чието натискане се тестват операциите с рационални числа: събиране, изваждане, умножение, деление; добавяне на рационално число към масива и сортиране на масива от рационални числа във възходящ ред;
 - г) етикет за извеждане на резултата от операциите.
 2. В метода **init** добавете слушатели за събитието „натискане на бутон“ към всички бутони. За слушател използвайте самия аplet.
 3. Напишете метод **getRationalNumber**, който връща рационално число и хвърля изключение **Exception**. Методът създава рационално число, като взема въведените стойности от текстовите полета. Хвърля следните изключения:
 - а) **NullPointerException**, ако съдържанието в някое от текстовите полета е празно;

- б) **NumberFormatException**, ако низът в някое от текстовите полета не може да се преобразува в **int**;
 - в) **RationalNumberException**, ако знаменателят на рационалното число е нула или и числителят и знаменателят са отрицателни числа (приемете, че числителят носи знака на рационалното число).
4. Реализирайте метода **actionPerformed**, който се извиква при вдигане на събитието **ActionEvent** „натискане на бутон“. Реализирайте всички операции за рационални числа, добавяне на рационално число към масива. За сортирането на рационалните числа използвайте метода **sort** на класа **Arrays (java.util)**
- ```
public static void sort(Object[] a)
```
- който сортира масива **a** във възходящ ред според естественото подреждане на неговите елементи. Елементите на масива трябва да реализират интерфейса **Comparable**.
5. Компилирайте аплета, променете кода в **\*.html** за влягане на аплета и го стартирайте в Microsoft Internet Explorer.