BIRZEIT UNIVERSITY

COMP231 – Advanced Programming

Assignment # 3

Objectives:

1. To apply class abstraction to develop software
2. To store, retrieve, and manipulate objects in an **ArrayList.**
3. To write a **try-catch** block to handle exceptions.
4. To use the **String** class to process immutable strings.
5. To read data from a file using the **Scanner** class.
6. To define a natural order using the **Comparable** interface.

Specification

Submission: Online **through Ritaj**.

What to submit: Your **own** **well-structured** and **well-commented** JAVA files (**.java**) (compressed into a **studentId_LabSec#.rar** file, e.g. **1234567_LabSec1.rar**).

Deadline: **23/4/2019** by midnight. (The online submission will be disabled after this time).

Task 1: Baby name popularity ranking

The popularity ranking of baby names in USA from years 2000 to 2017 is **attached**. Each line in each file contains baby's name, gender, and the total number for this name in that year.
Note: names are unsorted. To find the rank you have to sort based on number in descending order.
For example, in the file ***USA_yob2017.txt***, the female baby's name **Emma** are ranked #1 as it has the highest number **19738** between females.
Write a program that prompts the user to enter a valid **year**, **gender**, and followed by a **name**, and displays the ranking of the name for the year (if the name is listed). Here is a sample run:

Sample run1:

```
Please enter a year between 2000 and 2017: 1976
Please enter a year between 2000 and 2017: 2010
Please enter the baby's gender (M/F): m
Please enter the baby's name: Mamoun
Mamoun is ranked 11662
```

Sample run2:

```
Please enter a year between 2000 and 2017: 2010
Please enter the baby's gender (M/F): f
Please enter the baby's name: Abbas
Abbas is not listed in the file!!!
```

Task 2: Baby Name for Both Genders

Write a program that prompts the user to enter a valid year as in **task 1** and displays the names that are used for both genders in the file. Here is a sample run:

```
Please enter a year between 2000 and 2017: 2017
Liam,M,18728 <==> Liam,F,36
Noah,M,18326 <==> Noah,F,170
William,M,14904 <==> William,F,18
James,M,14232 <==> James,F,77
Logan,M,13974 <==> Logan,F,1103
Benjamin,M,13733 <==> Benjamin,F,8
Mason,M,13502 <==> Mason,F,58
Elijah,M,13268 <==> Elijah,F,26
```

Task 3: The Complex Class

A complex number is a number of the form $a + bi$, where **a** and **b** are real numbers and **i** is $\sqrt{-1}$. The numbers **a** and **b** are known as the real part and imaginary part of the complex number, respectively. You can perform addition, subtraction, multiplication, and division for complex numbers using the following formula:

$$a + bi + c + di = (a + c) + (b + d)i$$
$$a + bi - (c + di) = (a - c) + (b - d)i$$
$$(a + bi) * (c + di) = (ac - bd) + (bc + ad)i$$
$$(a + bi)/(c + di) = (ac + bd)/(c^2 + d^2) + (bc - ad)i/(c^2 + d^2)$$

You can also obtain the absolute value for a complex number using the following formula:

$$|a + bi| = \sqrt{a^2 + b^2}$$

Design a class named **Complex** for representing complex numbers and the methods **add**, **subtract**, **multiply**, **divide**, **abs** for performing complex-number operations, and override **toString** method for returning a **string** representation for a complex number. The toString method returns **a + bi** as a string. If **b** is **0**, it simply returns **a**.

Provide three constructors **Complex(a, b)**, **Complex(a)**, and **Complex()**. Complex() creates a Complex object for number **0** and Complex(a) creates a Complex object with **0** for **b**. Also provide the **getRealPart()** and **getImaginaryPart()** methods for returning the real and imaginary part of the complex number, respectively.

Your Complex class should also implement the ***Cloneable*** interface.

Write a test program that prompts the user to enter two complex numbers and display the result of their addition, subtraction, multiplication, and division. Here is a sample run:

```
Enter the first complex number: 3.5 5.5
Enter the second complex number: -3.5 1
(3.5 + 5.5i) + (-3.5 + 1.0i) = 0.0 + 6.5i
(3.5 + 5.5i) - (-3.5 + 1.0i) = 7.0 + 4.5i
(3.5 + 5.5i) * (-3.5 + 1.0i) = -17.75 + -15.75i
(3.5 + 5.5i) / (-3.5 + 1.0i) = -0.5094 + -1.7i
|3.5 + 5.5i| = 6.519202405202649
```

Good Luck!