TP3 on Java Programming

1 Introduction

The objective of the first lab is to practice implementation of classes and objects, the implementation of arrays.

The requirement of your report:

- Submit everything in a .zip file named: `JAVA_TP3_prenom_nom.zip`;
- Include a .pdf file to answer open questions;
- Name your project folder `TP3_Prenom_Nom`, and inside this project folder, include the following:
  - The following files: `.idea`, `.gitignore`, `.iml`, `out`, and `src` (your code should be inside the `src` folder);
  - Write comments in your code specifying which question you are answering;
  - Inside the `src` folder, include:
    1. A package called `arrayMuni`, containing the following classes [10%]:
       1) `ArrayMuni.java`
    2. A package called `contact` [50%], you can try your own structure design in order to finish it.
    3. A package called `Bonus`, containing the following classes [40%] [a]:
       1) `MonthDays.java` – Exercise 1;
       2) `ArraySum.java` –Exercise 2;
       3) `ArrayMaxMin` – Exercise 3;
       4) `RangeScore.java` – Exercise 4, and 5;
       5) `CinemaTicket.java` – Exercise 6.

Important: Write a report and submit it with code. The deadline of group1 is: 12:00:00, 05/11/2024; the deadline of group2 is: 12:00:00, 19/10/2024. Report should contains a pdf file with summary on what you have done and screenshot of codes. Zip of code should include multiple .java class files.

[a] these exercises are in Lecture 3 at `https://www.qiongliu.info/assets/teaching/java/Java_cm3.pdf`

2 Arrays

Create a class `ArrayMuni`, it has the constructor that creates 1-dimension array with length n.

1. It should have methods:
   - `Geti()`, which returns i-th element in the array.
   - `Seti()`, which returns i-th element in the array.
   - With error handling, by using `ArrayIndexOutOfBoundsException`.
   - `toString()`, which returns all value of elements in the array.
   - `Subarray()`: it returns a continuous subarray[1] with the maximum sum (the subarray contains at least one element).

[1] Examples of a continuous subarray A_1 from $A = \{a_1, a_2, a_3, a_4\}$, can be $A_1 = \{a_1\}$ or $A_1 = \{a_2, a_3\}$, or $A_1 = \{a_1, a_2, a_3, a_4\}$

2. In the `TestArray` file, create an instance from class `ArrayMuni`, and check the output of each method.
   - `MyArr1 =` {`-3,5,-3,6,-2,4,11,-5,4`}.
   - Display the message by calling method Geti(i), where i is the position of element in `MyArr1`, $i = \{0, 2, 10\}$.
   - `MyArr2 =` {`1,1,1,1,1,1,1,1,1,1`}
   - `MyArr3 =` {`5,4,-1,7,8`}
   - `MyArr4` has length n=20, each element is an integer, which is randomly generated with value between -50 and 50.

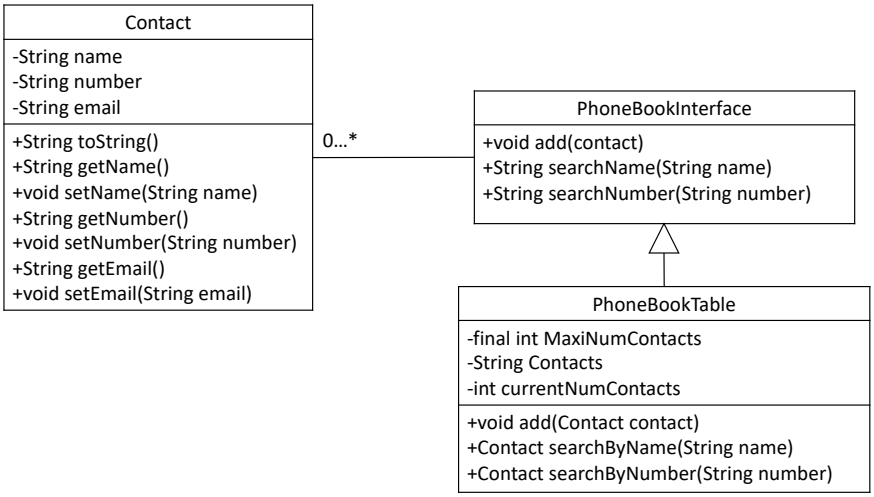
Questions:

1. Explain what how you design the algorithm to find optimal subarray with maximum sum.
2. How many combinations of subarrays did you find for a 1-dimension array of length n?
3. Is the algorithm you used, the simplest ways (in terms of complexity) to achieve the result?

3 Contact

We would like to create a simple phonebook system. First it contains the ability of creating new contact, with name, phone number and email. Then, the newly created contact is added to the phonebook. Phonebook should have a maximum capacity of m = 100 contacts.

Read the following class diagram, and write the Java codes to realize it.



Questions:

1. Write down the description you read from the class diagram, as well as the relationships between classes.
2. Create new `Contact` using different ways, either defining it directly in the test file, or by `Scanner`.
3. Use `(if {}else {throw})` for Exceptions when the new contact already exists.
4. Create the test file, and add following new contacts.
    - Jean, 0612121212, jean@gmail.com
    - Paul, 0618181818, paul@yahoo.com
    - Luc, 0646985221, luc@msn.com
    - Jean, 0612121212, jean@gmail.com

Important: Write a report and submit it with code before the end of the lab. Report should contains a pdf file with summary on what you have done and screenshot of codes. Zip of code should include multiple .java class files.