Gebze Technical University
Computer Engineering Department

CSE443 - Object Oriented Analysis and Design
Fall 2018-2019

Homework 3

Rule 1: no plagiarism (from colleagues or other sources). Detected cases of plagiarism will lead to a significant penalty of your course grade at the end of the semester.

Rule 2: no late submissions! Even if it is late by one minute, it will be ignored. Learning to plan your schedule according to deadlines is part of your education and an invaluable professional asset.

What to submit: a) the source code of your project *fully documented (with javadoc)*, b) a nicely formatted pdf report of your design decision explanations and class diagrams and c) an executable demo that fully illustrates your program's capabilities *whenever code is requested*.

****

Question 1 (20 points): After long hours at work you have finally completed the new java interface for credit card payments:

```
public interface ModernPayment{
        int pay(String cardNo, float amount, String destination, String installments);
}
```

However your company is still using extensively an old binary library from the 1980s called "TurboPayment" for card payments that it cannot afford to replace. "TurboPayment"'s interface looks like this:

```
public interface TurboPayment{
        int payInTurbo(String turboCardNo, float turboAmount,
                        String destinationTurboOfCourse, String installmentsButInTurbo);
}
```

Where all the method parameters have the same meaning and role as in ModernPayment.

Implement in Java a design pattern so that you can continue using all the classes implementing the TurboPayment interface with your new ModernPayment interface. Don't forget that the old library is binary, so you cannot modify the interface TurboPayment or the classes that implement it.

Question 2 (25 points): You are to design and implement an email address book.
Requirements:
a) Every email address must be stored along with the name of its owner.
b) An email address can be personal, such as alivelioglu@xyz.com or it can belong to a group, such as cengstudents4th@gtu.edu.tr which contains an arbitrary number of personal or group addresses.
c) Your application must be able to print on screen all the addresses in the address book. Personal addresses must be printed as : "alivelioglu@xyz.com Ali Velioglu" while group addresses must be printed by printing on screen every email (personal or group) address that they contain.
d) **Implement email addresses through the Composite design pattern.**
e) Draw a detailed class diagram of your solution.

Question 3 (25 points): It's the year 2019, and the Göktürk 3 satellite is in orbit. You work as a data analyst for the Turkish Space Agency. The satellite transmits data as 2D arrays of integers. You are tasked with coding two iterator classes for these data: one iterator that will print a 2D array spirally clockwise and another one that will print it spirally anti-clockwise, both starting at the top left element.

Example: if the data is

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

the first iterator should iterate it in the order: 1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10 and the second iterator as 1 5 9 13 14 15 16 12 8 4 3 2 6 10 11 7. Code in java **one of these iterators**, and provide a main class showing it in action. Do not implement the remove method of the iterator.

Question 4 (30 points): Both the 1D Discrete Fourier Transform and the 1D Discrete Cosine Transform do the same thing. Given a finite sequence of numbers, they express them us a sum of basis functions. Their difference being that DFT uses complex exponentials and DCT uses real-valued cosine functions. They are both very powerful transformations with numerous actual applications; such as jpeg compression, antenna technologies, etc.

TL;DR: given an array of numbers of length N, DFT and DCT both produce a new array of numbers also of length N. DFT produces complex numbers, DCT produces real numbers.

There is an abundance of online resources on both of them, explaining their calculation step by step.
https://en.wikipedia.org/wiki/Discrete_Fourier_transform
https://en.wikipedia.org/wiki/Discrete_cosine_transform – DCT II

Both transforms follow pretty much the same main procedure:

a) Read N numbers from a file
b) Transform the numbers into N outputs (DFT or DCT outputs..)
c) Write the N outputs to a new file
d) only in the case of DFT, **sometimes** the user wants additionally the time of execution printed on screen; by default no.

Implement both 1D transforms, and make sure that you apply the **Template Method design pattern**.

Good luck.