Due Date: 20.12.2020 23:55

CENG211 – Programming Fundamentals
Homework #3

In this homework, you are expected to implement a simplified version of the "Cargo Acceptance Application" in Java.
You should fulfill the concepts of:

- Abstract Classes
- Interfaces
- Enumeration
- Exception Handling
- Generics
- Collections

In this application, there are two different types of cargo packages: e-commerce cargo package and normal cargo package. For each cargo package, size is calculated as follow:

desi = (width*length*height)/3000

size=max(desi,weight)

**Please note that weight should be entered as kg and width, length, height should be entered as cm. The size value should be an integer.*

Normal cargo packages are sent by individual customers, and the cargo is not commercial. The customer pays for the cargo at the agency. The price is calculated based on the formula below:

price = 18,5+3*size

E-commerce cargo packages are sent by commercial customers who sell their products on e-commerce sites such as Amazon, Hepsiburada, N11, and Trendyol. The customer has a cargo code and submits this code to the agency for acceptance of the package. **(Price is not calculated.)** The code format differs for each e-commerce site (***Hint**: Generics*). Also, each agency has daily limits for each e-commerce site. The code format and daily limits are as follow:

Table 1. E-commerce Code Format and Daily Package Limits

E-commerce Site	**Code Format**	**Daily Package Limit**
Amazon	7 digit integer	5
Hepsiburada	String (length with 8 characters)	7
N11	String (length with 7 characters)	6
Trendyol	8 digit integer	9

The file that consists of different cargo packages information is given: HW3_PackagesToAccept.csv The columns of the file are given in the following for each different cargo package type:

Normal Cargo Package: ***type****, ***sender id (11 digits), sender name, recipient name, recipient address, weight, width, length, height***

E-commerce Cargo Package: ***type****, ***e-commerce site name, cargo code, weight, width, length, height***

**For normal cargo packages type is indicated as 'Normal' and for e-commerce cargo packages type is indicated as 'Ecommerce' in the given input file.*

The application should work as follow:

1. Read the files to process the cargo packages for acceptance.
    a. If *sender id* does not consist of 11 digits, an IDNotCorrectException should be thrown. (for normal cargo package)
    b. If *cargo code* format is not appropriate for e-commerce cargo packages (see Table 1), necessary exceptions should be thrown. (You should think about the exceptions.)
2. Calculate the predicted delivery day (i.e. Wednesday) of the cargo. The shipment duration is 2 days and there is no delivery on Sundays. The acceptance date is your system date. (If the system date is Monday, the predicted delivery day should be shown as of Wednesday) ***Hint:*** *Apply enumeration for the days.*
3. Calculate the price of normal cargo packages based on the given formula.
4. Generate a 7-digit integer cargo code for each accepted normal cargo package.
5. The application should check for the daily limits for each e-commerce site (see Table 1). If the limit is exceeded and its status should be set as *Not Accepted.* The status should be set as *Accepted* unless the limit is exceeded. ***Hint:*** *The status can be Accepted* or *Not Accepted.*
6. There is no daily limit for normal cargo packages.

Sample Output:

Welcome!

Number of Accepted Cargo: ***…***

Number of Not Accepted Cargo: ***…***

Here are the details:

Normal Cargo Packages:

No Cargo Code Sender ID Sender Name   Recipient Name Recipient Address Size Price Delivery Day

1   …….

2   …….

3   …….

..

E-commerce Cargo Packages:

No E-commerce Site Cargo Code Status Size Delivery Day

1   …….

2   …….

3   …….

4   …….

..

Hints on Interfaces:

In the future,

1. New cargo package (apart from normal and e-commerce cargo package) type may be added to the system.
2. The cargo company may define a discount for a group of customers. Thus, we may also need a discounted price calculator for normal cargo packages.
3. The company may contract with new e-commerce sites. The code format and the daily limit for the new e-commerce site may be different from the current sites.
4. The application may be requested to read different file types (such as JSON, XML, etc.).

**** In this assignment, you should achieve total abstraction and loose coupling by implementing the necessary interfaces.***

Important Notes:

1. You can use standard **java.io** packages to read files. Do NOT use other 3rd party libraries.

2. You should use relative paths (e.g. `Files/sample.csv`) instead of absolute paths (e.g. `C:\\user\\eclipse-workspace\\MyProject\\Files\\sample.csv`).

3. To support **Turkish characters** you may need to change your project's text file encoding to UTF8: Right-click on your project (in package explorer) → Properties → Text file encoding → Other → UTF8 → Apply.

4. You are expected to write clean, readable, and tester-friendly code. Please try to maximize reusability and prevent redundancy in your methods.

Assignment Rules:

1. In this lecture's homework, there is no cheating allowed. If any cheating has been detected, they will be graded as 0 and there will be no further discussion on this.
2. You are expected to submit your homework in groups. Therefore, only one of you will be sufficient to submit your homework.
3. Make sure you export your homework as an Eclipse project. You can use other IDEs as well, however, you must test if it is supported by Eclipse.
4. Submit your homework through CMS.
5. Export your Java Project with your assigned group ID (which will be announced on CMS) as the given format below:

   G05_CENG211_HW3.zip

6. Please be informed that your submissions may be anonymously used in software testing and maintenance research studies. Your names and student IDs will be replaced with non-identifying strings. If you do not want your submissions to be used in research studies, please inform the instructor (Dr. Tuglular) via e-mail.

Grading:

Item	**Marks (Total 100)**
Interfaces	30
Generics	10
Enumeration	10
Collections (Arraylist operations)	5
Exception Handling	10
Reading Data from Files	5
Data Classes (including Abstract Class)	15
Remaining Code	15