ECONOMICAL UNIVERSITY OF ATHENS

COMPUTER SCIENCE DEPARTMENT

Computer Programming withJava

Spring semester2019-20

3theWork

In3thetask you will extend the program you created to2thework.

Wanted:

A)To expand its program2hertask, so that it reads the list of devices available for sale, the list of orders and the list of sales from structured text files. There should be three different files for the three directories. The structure (format) of the files is given in Appendices 1, 2 and 3. H reading to be done at the start of the program. When reading, a check should be made for the correct opening of the file and the correct syntax of the data. The bound words (tags) in files must be recognized in both upper and lower case.

B)To expand its program2hertask so that it writes the order list and sales list data to structured text files. The structure (format) of files are the same as in Question A, and are given in the Appendices2 and 3. The same files with the list of orders and the list of sales that you opened at the start of the program should be updated with all changes (new orders and sales, updating the number of available pieces of a device) that have been made during the execution of the program. Registration should be done with the command to terminate the program. If you want, you can make intermediate recordings while the program is running.

The available for sale, order list, and sales list files can be created initially with a text editor.

Number of units3herwork: 0.6 / 2 (total assignments participate in the final grade with rate20%)

Date of announcement: May 25, 2020

Delivery date: June 9, 2020, time 23:59

Delivery method: Online submission viae-class

Annex1– Structure of the archive of devices available for sale

Spaces can be of any length and consist of either “ ” characters (spaces) either by characterstabs (“/t”). All entities start and end with “{” and “}” respectively, on a separate line.

Inside the entity declared with the bound word (tag) ITEM must exist o device codeCODE, the type of device ITEM_TYPE (TV, refrigerator, etc.), the modelMODEL, the model year MODEL_YEAR, the manufacturer MANUFACTURER, the pricePRICE, as well as the other characteristics of the specific type of device. For these additional features define appropriate reserved words (tags). Invalid features, i.e. features that are not available/ recognized for the specific device type should be ignored. Also, blank lines should be ignored.

In aITEM_LIST (device list) can include multiple entities ITEM. Reviewing the work may modify your file to test your code's fault tolerance of the file structure.

The sequence of bound words (tags) must not be unique. That is, the code, its type device, manufacturer, etc., can appear in any order. When someonetags does not exist, the corresponding attribute should be given a default value. The fieldsCODE, ITEM_TYPE and MODEL must be present. If there are none, the specific object to be ignored and a relevant notification printed on the screen. Note that in order to instantiate and initialize the appropriate class, it must first be readtag ITEM_TYPE. However, the specific tag may not is the first one encountered within the ITEM entity.

```
ITEM_LIST
{
    ITEM
    {
        CODE ABF010
        ITEM_TV TYPE
        MODEL 27UL600-W
        MODEL_YEAR 2019
        MANUFACTURER LG
        PRICE 450
        PANEL_TYPE IPS
        DIMENSIONS 27
        RESOLUTION “Ultra HD 4K” INTERFACES “2 X
        HDMI / 1 X DisplayPort” PIECES 2

        . . .
    }
    ITEM
    {
        CODE CDA015
        ITEM_TYPE refrigerator
        MODEL KNT56AW20
        MODEL_YEAR 2019
        MANUFACTURER Pitsos
        PRICE 930
        PIECES 3
        . . .
    }
    . . .
}
```

Annex2– Structure of the order file

Spaces can be as long as you like and consist of either “ ” characters (spaces) either by characterstabs (“/t”). All entities start and end with “{” and “}” respectively, on a separate line.

Inside the entity declared with the bound word (tag) ORDER you must put all the characteristics of the specific order. Inside the entityORDER_LIST (orders, only one such entity is declared in your file) multiple entities can be includedORDER.

When reading the orders, the program should check that the orders are for devices that are in the list of available devices. If an order concerns a device that is not included in the list of available devices, it will be ignored and a relevant notification will be printed on the screen. The order oftags don't it must matter.

```
ORDER_LIST
{
    ORDER
     {
        ITEM_TV TYPE
        MODEL 27UL600-W
        MANUFACTURER LG
        ORDER_NUMBER 1001
        NAME “Giorgos Kalos”
        ORDER_DATE “6 May 2020”
        DELIVERY_DATE “25 June 2020”
        PRICE 450
        STATUS EXPECTED
        . . .
      }
       . . .
}
```

Annex3– Structure of the sales file

Spaces can be as long as you like and consist of either “ ” characters (spaces) either by characterstabs (“/t”). All entities start and end with “{” and “}” respectively, on a separate line.

Inside the entity declared with the bound word (tag) SALE you must put all the characteristics of the particular sale. Inside the entitySALES_LIST (sales, only one such entity is declared in your file) multiple entities can be includedSALE.

When reading the sales, the program should check that the sales are for devices listed in the available devices list. If a sale concerns a device that is not included in the list of available devices, it should be ignored and a relevant notification will be printed on the screen. The order oftags should not to matter.

```
SALES_LIST
{
        SALE
        {
             ITEM_TV TYPE
             MODEL 27UL600-W
             MANUFACTURER LG
             SALES_NUMBER 2001
             NAME “Anesths Kakos”
             DATE "January 29, 2020". . .

          }
           . . .
}
```