Middle East Technical University Department of Computer Engineering

CENG 443

Introduction to Object Oriented Programming and Systems

Fall 2019-2020

Homework 3 - PC Parts System

Due date: December 30th, 2019, Monday, 23:55

1 Introduction

This assignment aims to familiarize you with Java 8 Streams (not to be confused with Java I/O Streams). You are hired by a PC parts store and asked to implement a query system in order to monitor their inventory. In this assignment you will use Java 8 Streams and lambda functions.

2 Problem Definition

The PC Parts store has given you a csv file of their database[1]. Application you will create will use this file as the data source and provide 5 methods to query/process this information.All parts in the store have a `Type`, a `Brand`, a `Model` and a `Price` associated with them. The type of a part defines the specific properties the part has. For example parts with the type `Monitor` have an aspect ratio property while parts with type `CPU` have a core count property. Each part's type can be one of the following: `Hard Drive`,`Monitor`,`PSU`,`Motherboard`,`CPU`,`GPU`,`Memory`,`Mouse`,`Keyboard`. Specific properties of different types are further explained in the CSV File Specifications section.The `Brand`,`Model` and `Price` are self explanatory. These properties don't have a special meaning like the `Type`,they are just needed to further describe a part. A point to note about the `Price` field is that it can be set to 0 dollars. This means that the item is currently not available at the store.

2.1 CSV File Specifications

Each line in the csv file corresponds to an item. Each line starts with the type of the item and are followed bythe properties associated with the type. Properties are seperated from each other by a single comma and nothing else. Properties for each type are listed below.

- Hard Drive : `Type`, `Brand`, `Model`, `Capacity` (in GBs), `Price`
- Monitor: `Type`, `Brand`, `Model`, `Aspect Ratio`, `Size` (in inches), `Price`
- PSU: `Type`, `Brand`, `Model`, `Form Factor`, `Wattage`, `Price`
- Motherboard: `Type`, `Brand`, `Model`, `Socket`, `RAM Slots`, `Price`
- CPU: `Type`, `Brand`, `Model`, `Core Count`, `Clock Speed` (in GHz), `Price`

[1]All parts, their specs. and prices are taken from pcpartpicker.com

- GPU: `Type`, `Brand`, `Model`, `Chipset`, `Capacity` (in GBs), `Clock Speed` (in MHz), `Price`
- Memory: `Type`, `Brand`, `Model`, `Socket`, `Capacity` (in GBs), `Clock Speed` (in MHz), `Price`
- Mouse: `Type`, `Brand`, `Model`, `Connection type`, `Price`
- Keyboard: `Type`, `Brand`, `Model`, `Connection type`, `Price`

Properties of each type are ordered in the csv file in this order:

```
Hard Drive,Brand,Model,Capacity,Price
Monitor,Brand,Model,Aspect Ratio,Size,Price
PSU,Brand,Model,Form Factor,Wattage,Price
Motherboard,Brand,Model,Socket,RAM Slots,Price
CPU,Brand,Model,Core Count,Clock Speed,Price
GPU,Brand,Model,Chipset,Capacity,Clock Speed,Price
Mouse,Brand,Model,Connection type,Price
Memory,Brand,Model,Socket,Capacity,Clock Speed,Price
Keyboard,Brand,Model,Wired,Price
```

So with actual values it looks like this:

```
Hard Drive,ADATA,Premier,960GB,129.98 USD
Monitor,Acer,V226HQL Abmid,16:9,21.5,102.99 USD
PSU,Cooler Master,MWE Bronze V2,ATX,550,0.00 USD
Motherboard,ASRock,FM2A68M-HD+,FM2+,2,53.98 USD
CPU,Intel,Xeon E5-2670 V3,12,2.3GHz,694.56 USD
GPU,Asus,GeForce RTX 2080,Turbo,8GB,1515MHz,819.95 USD
Mouse,Logitech,M325c Urban Grey,Wireless,25.36 USD
Memory,G.Skill,Trident Z Royal 64 GB,DDR4,8GB,3200MHz,838.99 USD
Keyboard,MSI,Vigor GK30,Wired,49.99 USD
```

2.2 Implementation Specifications

You are asked to implement 5 methods that queries or processes the parts in the csv file. These methods should be implemented in the class `PartsStore` which is provided to you.

- `FindPartsWithBrand(String type, String brand)`: Prints all the parts of `type` with `brand`. If the `type` is `null` it returns all items with brand regardless of their types.
- `TotalPrice(String type, String brand, String model)`: Prints the total price of the parts with `type`, `brand` and `model`. In case of one or many arguments being `null`, total price of all parts with the not null arguments are returned. For example, `TotalPrice(null, "Asus", null)` will return total price of all parts with the brand `"Asus"`.
- `UpdateStock()`: Discards the parts that are not in the stock right now (Parts with price value set to 0 USD) and prints how many items are discarded. After this method is called, other methods should work over the updated stock, so that they would never return an item with price 0 USD.
- `FindCheapestMemory(int capacity)`: Prints the details of the cheapest Memory part with equal or larger capacity than `capacity`.
- `FindFastestCPU()`: Prints the details of the CPU with the highest value of `(core count * clock speed)`

3 Input & Output

3.1 Input

You won't be dealing with any input directly in this assignment. Instead we will use the class(es) you implemented during evalution. The methods you have implemented will be called from the main function in a similar fashion to the snippet below.

```
PartsStore ps = new PartsStore();
ps.FindPartsWithBrand("Keyboard", "Logitech");
ps.FindFastestCPU();
ps.UpdateStock();
ps.FindFastestCPU();    // Output of this call may be different than the
                        // previous one due to previous UpdateStock call.
```

Plase make sure that the method calls to `PartsStore` class works as expected.

3.2 Output

Each of the 5 methods you will implement will print their results to the `stdout`. When the output of a method is a part or a part list, each part should be printed on a separate line and the properties of the part should exactly be written as seen on the csv. Examples for output formats of methods:

```
>> FindPartsWithBrand("GPU", "Asus"):

        ...
        GPU,Asus,GeForce RTX 2080,Turbo,8GB,1515MHz,819.95 USD
        GPU,Asus,GeForce RTX 2080 Ti,Turbo,11GB,1350MHz,1139.99 USD
        GPU,Asus,GeForce GTX 1080 Ti,11GB,1480MHz,1145.99 USD
        ...


>> TotalPrice("GPU", "Asus", "GeForce RXT 2080"):

        7754.52 USD


>> UpdateStock():

        1302 items removed.


>> FindCheapestMemory(16):

        Memory,Corsair,Vengeance RGB Pro 64 GB,DDR4,16GB,2666MHz,299.99 USD


>> FindFastestCPU():

        CPU,AMD,Threadripper 2990WX,32,3GHz,0.00 USD
```

Note that these outputs are just an example, it may not be the correct output. They are provided as a concrete example for the output formats of the methods. Also the output order of `FindPartsWithBrand()` doesn't matter.

4 Specifications

- Your code must be written in Java
- You must use Java 8 Streams and Lamda Functions where possible. Try to utilize streams as much as youcan, efficiency of your programs is not important as long as they finish execution in a reasonable amount of time.
- Submissions will be evaluated with both black box and white box techniques. Correctness of your outputs isimportant. White box evaluation will be employed to make sure that your code adheres to OOP principles.
- Non-terminating submissions will suffer a penalty, as well as submission that don't adhere to OOP principles.
- Everything you submit should be your own work. Usage of binary source files and codes found on internet is strictly forbidden.
- Please follow the course page on ODTUClass for updates and clarifications.
- Please ask your questions related to homework through ODTUClass instead of emailing directly to teachingassistant.

5 Submission

Submission will be done via ODTUClass. Create a zip file named `hw3.zip` that contains all your java source code files without any directory. Make sure that your `PartsStore` is in this archive. Your code should be able to compile and run using this command sequence after being unzipped.

```
> javac *.java
```