Basic Programming Exam

Prof. Dr. Guy Baele
1pm, January 11th, 2021, QDV 01.100 / 200C 01.01 / 200C 01.03

Instructions

- There is only one question due to the time restriction of 1.5 hours (students with facilities are allowed to work for 2 hours).

- Assure yourself that there 6 pages in this Word document (the exam question + the provided code + target output of the program).

- To hand in, create a .zip file of all your Java files named "Surname_Firstname.zip" (e.g. "Baele_Guy.zip"). Do not submit .jar files! Only include .java files; there is no need to create separate packages in your code (i.e., keep all your .java files in the src folder). There is no need to write a report but adding comments to your code is encouraged.

- The programming exam lasts for 1.5 hours at the most (the R part starts after you hand in), and for 2 hours if you are permitted to work for longer. The Toledo system will not close automatically, but we will monitor the time at which you submit your solution, and your submission time will be used to determine whether your solution was submitted on time; you can submit as many times as you want, but we will only correct your final submission. **Only submissions through Toledo are accepted! Late submissions will not be graded! Submissions via e-mail will not be graded!**

- Be sure to use the provided file(s) for the questions from this document, which will be made available on Toledo during the length of the exam (i.e., don't manually type out all of the provided code in this document!). Pay attention to encapsulation! As with all parts of this course, the key is to focus on principles of object-oriented programming, and hence not merely that "the program works".

- The code you are provided comes from a working implementation and as such does not contain any errors.

- Don't forget to sign your name on the sheet near the exit of the classroom before you leave.

Question: Object-oriented Programming

In this question/exam, the goal is to write the required classes to make the code below run in its entirety and without error, as well as to produce the provided output below. To this end, you receive a StoreManager.java file of which **you may not alter the main(String[] args) method!** Adding methods and/or variables to this .java file is not permitted.

An electronics Store has an inventory of portable devices – such as laptops and tablets – and other devices (non-portable ones such as desktop computers). Here, you need to create an inventory system for the store and design the classes required to create objects of two types of portable devices: Laptop and Tablet, as well as one other device: DesktopComputer.

A Laptop is defined by the following properties (provided as arguments to its constructor in the following order):

- Short summary information, typically consisting of brand and model number
- Brand of the laptop
- Its weight
- Its screen diagonal
- A list of its wireless connections
- The material from which the chassis is made
- Whether or not it comes with a separate mouse
- Whether or not it comes with a separate keyboard

A Tablet is defined by the following properties (provided as arguments to its constructor in the following order):

- Short summary information, typically consisting of brand and model number
- Brand of the tablet
- Its weight
- Its screen diagonal
- A list of its wireless connections
- The material from which the chassis is made
- Whether or not it comes with a stylus
- How much its accompanying case weighs

A DesktopComputer is defined by the following properties (provided as arguments to its constructor in the following order):

- Short summary information, typically consisting of brand and model number
- Brand of the computer
- Number of processor cores
- Amount of memory
- Its screen diagonal
- Number of hard drives

Important: all portable devices (laptops and tablets, but also other types of portable devices that could be added to the program in the future) must implement the following two methods:

- public double getWeight();
- public ArrayList<String> getWirelessConnections();

For tablets, the weight equals the total weight of the tablet and its case, whereas for laptops the weight simply equals the weight of the laptop itself.

There is no need to implement functionality that enables to buy (portable) devices from the store; you should focus on generating the code for the required classes (you need to determine for yourself which and how many classes are required) and generating the output to screen.

Provided .java file (see also the provided file on Toledo):

```
import java.util.ArrayList;

public class StoreManager {

       public static void main(String[] args) {

             Store localStore = new Store("Electronics Inc.");

             //create desktop computers
             DesktopComputer modelOne = new DesktopComputer("Acer Aspire TC-895 
I7510", "Acer", 8, 16, 24, 1);
             DesktopComputer modelTwo = new DesktopComputer("MSI MEG Infinite X 
10SD-661MYS", "MSI", 8, 16, 27, 2);
             DesktopComputer modelThree = new DesktopComputer("Apple iMac 27 inch 
Intel Core i7", "Apple", 18, 32, 27, 1);

             //add desktop computers to store
             ArrayList<Device> desktopList = new ArrayList<Device>();
             desktopList.add(modelTwo);
             desktopList.add(modelThree);

             localStore.addDevice(modelOne);
             //add 5 more modelOne devices to the inventory
             localStore.addDevice(modelOne, 5);
             localStore.addDevice(desktopList);

             //create laptop connections
             ArrayList<String> laptopConnections = new ArrayList<String>();
             laptopConnections.add("Wifi");
             laptopConnections.add("Bluetooth");
             //create laptops
             Laptop laptopModelOne = new Laptop("Apple MacBook Pro 16 inch Touch 
Bar Silver", "Apple", 2.0, 16.0, laptopConnections, "aluminum", false, false);
             Laptop laptopModelTwo = new Laptop("HP 14s-dq1031nb", "Hewlett 
Packard", 2.78, 17.3, laptopConnections, "plastic", false, false);
             Laptop laptopModelThree = new Laptop("Lenovo IdeaPad L340-17IRH 
81LL003EMB", "Lenovo", 1.46, 14.0, laptopConnections, "plastic", false, false);

             //create laptop connections
             ArrayList<String> tabletConnections = new ArrayList<String>();
             tabletConnections.add("Wifi");
             tabletConnections.add("Bluetooth");
             tabletConnections.add("5G");
             //create tablets
             Tablet tabletModelOne = new Tablet("Samsung Galaxy Tab A7 64GB 
Wifi", "Samsung", 0.472, 10.4, tabletConnections, false, 0.505);
             Tablet tabletModelTwo = new Tablet("Apple iPad Pro 12.9 inch 128 GB 
Wifi Space Gray + Pencil 2", "Apple", 0.641, 12.9, tabletConnections, true, 
0.349);

             //add laptops and tablets to store
             ArrayList<PortableDevice> portableDevices = new 
ArrayList<PortableDevice>();
             portableDevices.add(laptopModelOne);
             portableDevices.add(tabletModelOne);

             localStore.addPortableDevice(portableDevices);
```

```
        localStore.addPortableDevice(laptopModelTwo);
        //add 5 laptopModelThree portable devices to the inventory
        localStore.addPortableDevice(laptopModelThree, 5);
        //add 1 tabletModelTwo device to the inventory
        localStore.addPortableDevice(tabletModelTwo, 1);

        //print the inventory of the store
        System.out.println(localStore);

        //print total number of items in store
        System.out.println("Number of devices in store: " + 
localStore.getNumberOfDevices() + "\n");

        //print all Apple device models from store
        ArrayList<String> result = localStore.retrieveDevices("Apple");
        System.out.println("Apple devices in store:");
        for (String item : result) {
            System.out.println(item);
        }
        System.out.println();

        //clear out the entire inventory
        localStore.clearInventory();

        //print the inventory of the store
        System.out.println(localStore);

    }

}
```

Required output from running the provided main method:

```
Electronics Inc.
Devices in stock:
  Apple iMac 27 inch Intel Core i7: 1
  Acer Aspire TC-895 I7510: 6
  MSI MEG Infinite X 10SD-661MYS: 1
Portable devices in stock:
  Apple MacBook Pro 16 inch Touch Bar Silver: 1
  Samsung Galaxy Tab A7 64GB Wifi: 1
  HP 14s-dq1031nb: 1
  Lenovo IdeaPad L340-17IRH 81LL003EMB: 5
  Apple iPad Pro 12.9 inch 128 GB Wifi Space Gray + Pencil 2: 1

Number of devices in store: 17

Apple devices in store:
Apple iMac 27 inch Intel Core i7
Apple MacBook Pro 16 inch Touch Bar Silver
Apple iPad Pro 12.9 inch 128 GB Wifi Space Gray + Pencil 2

Electronics Inc.
Devices in stock:
none
Portable devices in stock:
none
```