

CS1027b Computer Science Fundamentals II

Assignment 1

Description

The PooPoo Conglomerate of Canada has decided to enter into the magazine subscription service business in Canada. To this end, they need to build a customer billing service for magazine subscribers as a Java program. Magazines are available in both digital and print formats at different prices. In this assignment, you are required to develop a mini-version of such a system. Given a list of customers and their magazine subscription information and a list of magazines and their prices you are to generate bills for each customer.

Due Sunday January 25th via Owl.

Functional Specifications

Create a class `Magazine` that has 3 private variables, `magazineName`, `magazineFormat`. Code three getter methods, `getMagazineName`, `getMagazineFormat` and `getMagazinePrice`. A single constructor class `Magazine` creates and objects with three parameters corresponding to the three private variables.

Create a class `MagazineCollection` that has a private variable that is an array of type `Magazine` and stores our collection of magazine objects. Another private variable is `numberOfMagazine`, which is an integer and gives the total number of magazines PooPoo sells in all formats.

These arrays are allocated by a constructor method, also named `MagazineCollection`, which creates an object with an empty array and sets `numberOfMagazine` to 0. The constructor is overloaded. If is invoked with no parameters, then it uses a default array size of 5. If it has a positive integer as its single parameter, then arrays of that size are allocated. However, make this smaller than the number of magazines so the TA can see that your helper method `expandCapacity` (below) works.

A second method, `addMagazine`, in `MagazineCollection` is used to populate these arrays. As in the `SocialNetwork` given in class, use a helper method, `expandCapacity`, to expand the array if needed.

A third method in `MagazineCollection`, `searchMagazinePrice`, has two formal parameters, `name` and `format`, and searches the array for entries equal to those parameters and returns the price of that magazine (as a double).

Of course, there is a `toString` method in `MagazineCollection` that can print the magazine information to a string. Lastly, there is a getter method, `getNumberMagazines` in `MagazineCollection`, that returns the number of magazines stored in a `magazineCollection` object.

Data for this assignment is provided in files [magazine.txt](#) and [customer.txt](#). The `magazine.txt` file has one line for each magazine, specifying the magazine's name, its format and its price. The `customer.txt` file contains information for each customer separated by blank lines [that means the last line of the file is also a blank line]. The first line contains the customer's first name, his/her last name and his/her customer number. The second and subsequent lines contain the subscription names (a string for a magazine name) and subscription formats ("print" or "digital"). You may assume all the data is correct and there are no

errors possible in the bill calculations (so don't worry about Exceptions for this assignment).

You are provided a class [InStringFile](#) that allows you to read one line at a time from a data file, as a `String`. We also provide a partially finished method [Main](#) that uses `InStringFile` to read lines of text from files `magazine.txt` and `customer.txt`.

The `Main` class has a `public static void main(String[] args)` method that reads the `magazine.txt` and stores this information in a `Magazine` object. Then each customer in `customer.txt` is processed one at a time and a bill specifying the customer's subscription information is printed. This includes the cost of each subscription and the total cost. Customer bills should be "pretty" printed and separated by a blank line. As you can see in `Main.java` printing is done using formatted output via `System.out.format`.

The `Main.java` uses local variables for `customerNumber`, `customerFirstName` and `customerLastName` to read customer information. There are local variables for `magazineName` and `magazineFormat` which are read line by line from `customer.txt` and then processed. A line in the customer file has 3 tokens (customer first name, customer last name, customer number), has 2 tokens (magazine name and magazine format) or has 0 tokens (a blank line). This is used to distinguish between customer identification information and subscription information. This code is reasonably complete. You will need to add some code to `Main.java` at the locations indicated by comments to create a `Magazine` object and then to process the bills.

Non-functional Specifications

- Your program has to be compilable under Eclipse.
- Use Javadoc comments for each class and method. All significant variables must be commented.
- Use Java conventions and good Java programming techniques (meaningful variable names, conventions for variable and constant names, etc). Indent your code properly.
- Remember that assignments are to be done individually and must be your own work.