COMPSCI 2S03, Principles of Programming
Assignment 5, Fall 2019

Hassan Ashtiani, McMaster University

Due date: Sat, Nov 30, 11:59pm

- You should write your codes in Java. The input/output of the programs are all from/to standard-input/standard-output (i.e., reading from the keyboard, and writing to the text terminal).
- DOMJudge is used to automatically test your code against a number of testcases. To submit to DOMJudge, login with your username and password on http://130.113.70.122/domjudge/team from the campus (or via VPN) and then submit your source files for each question.
- In addition to DOMJudge, submit your source codes on Avenue. Upload a single zip file named `macID_assignment5.zip` (e.g., `adtdb12_assignment5.zip`). The zip file should include only one folder, called `adtdb12_assignment5`, and this folder should include all your .java source files (and nothing else). This makes marking much faster for the TAs, so **any violations from this will be penalized**.
- Your source codes will be graded by the TAs; so follow the style conventions for coding from the class (naming, indentation, spacing, comments) and good coding practise to get full marks.
- Use Piazza for discussions and clarifications about the assignment.
- This assignment has bonus points. If you get a grade above 100, then it can compensate for your other assignments (but not for the exams).

This assignment is a follow-up on Assignment 4. You do not need to have scored perfectly in the previous assignment to get a good mark on this one: solving this one is rather independent to your **solution** to the previous assignment (still, your knowledge of the problem will certainly be reused).

First of all, we have made some changes to the problem and we have updated the code accordingly. Here are the details:

- The description of each CAR now includes a PLATE_NUMBER which uniquely identifies that car (no two cars can have the same plate number). Furthermore, we assume that POSTAL_CODE uniquely identifies a home.
- A customer can now own multiple homes and multiple cars (which can be distinguished by their postal codes or plate numbers respectively).
- When a customer signs a contract (the CONTRACT block in the input) under a car plan, "all of her cars" will be insured under that plan. Also, when a customer signs a contract under a home plan, "all of her homes" will be insured under that plan. Just like before, we do not check any eligibility criteria at the time of signing a contract (we will do it only at the time of making a claim). The name of the contract is unique (no other contract can have the same name).
- Because each customer may own multiple cars/homes, for each claim we need to specify which insurable item we are talking about. Therefore, the CLAIM block now includes an INSURABLE_ID which is equal to the POSTAL_CODE for homes or PLATE_NUMBER for cars.

- The claim may be accepted/rejected based on the plan criteria (just like previous assignment). We also need to double-check the fact that the INSURABLE_ID is consistent with the type of the plan (e.g,. one cannot make a claim regarding a home under a car plan) and belongs to the customer that is making the claim. We have updated the processClaim function in BlockCommand class to reflect these changes.

- The PRINT PLAN command has been removed from the question (PRINT CUSTOMER is still there) to reduce the dependency between your score in this assignment and in the previous one.

The updated code has been provided with the assignment. You are strongly encouraged to use this new code (rather than assignment 4's) because of these updates. You can also take a look at the updated sample_input.txt.

Note: DOMJudge supports at most 20 source files. Therefore, we have merged Utils.java and Main.java in one files to reduce the number of source files to 19 (so you can add one more file). If you needed to add more than 1 class, then you are allowed to add them all in the same java file to avoid going beyond 20 files.

1. [**60 points: DOMJudge (30 points), code (30 points)**

   The following two issues were reported by the students for assignment 4. In this question you need to modify the code to fix both of the issues. Note that you **do not need to** add comments again to all the given files; you only need to add comments to the parts of the code that you would modify/add for this question.

   First issue. There was a bug in the previous assignment when we were checking whether the age criterion is met for a customer/home. By subtracting the birth/build year from the current year we got a rough estimate of the age but this may be inaccurate (e.g., if the person is born on Dec 28th 2000 and makes a claim on Jan 1st 2018 he is not 18 years old yet). Modify the code to fix this issue. For example, a person born on Dec 28th 2000 will become 18 years old on Dec 28th 2018; therefore, CUSTOMER.AGE $>$ 18 will be satisfied on Dec 28th 2018 or any dates afterwards. Also, CUSTOMER.AGE $<$ 18 will be satisfied for any date strictly before Dec 28th 2018.

   Second bug. There was another subtle bug in the previous assignment in the cases where we had multiple criteria like CUSTOMER.AGE $>$ 30 and CUSTOMER.AGE $<$ 60 for a plan. The problem was that both of these criteria/tags would have been stored in a single HashMap. Unfortunately, we would use the same key for these two entries (both keys would be "CUSTOMER.AGE"). Because HashMap does not support repetitive keys, we would actually override the first criterion when we are adding the second one to the HashMap. Modify the code to fix this issue to be able to support multiple criteria regarding the same variable.

2. [**60 points: DOMJudge (30 points), code (30 points)**

   In this question we want to add a new type of entity called a "Company". Take a look at sample_input2.txt and sample_output2.txt. (note that the given code does not generate the expected output because handling companies is not implemented there; that's what you need to do).

   A company is defined in a BEGIN_COMPANY...END_COMPANY block. It has an OWNER_NAME, a COMPANY_NAME (which is unique for among other companies) and a value. Note that the owner name is either a customer's name that was defined before, or another company's name.

   The wealth of a customer is defined by the sum of the values of the cars, homes, and companies she owns. Note that some of the companies may be owned by a customer indirectly. For example, in the given sample_input2.txt, we see that David owns Googoogle which is valued 9000000. Furthermore, DeepThoughts is owned by Googoogle so it is owned by David as well. In this case, and given the fact that David owns a home and a car as well, the total wealth of David is 9000000+1000000+2000000+100000=12100000 (where 2000000 is for his home and 100000 is for his car).

Now what is the use of this notion of wealth? In the definition of a **car plan or a home plan**, we can have a new criterion called CUSTOMER.WEALTH. Therefore, the eligibility of a customer can be dependent on his wealth. You need to modify the code to support this feature.