

Persistent Data Strategy for the Finance Application

Model Classes:

- Asset (assetName, assetType, estimatedYearlyIncome, incomeRate)
- Employee (name, birthDate, age, jobName, jobHoursWorked, hourlyPay)
- Liability (liabilityName, liabilityType, estimatedYearlyInterest, interestRate)
- Bill (super(), billFrequency, billCost)
- Loan (super(), loanAmount)
- Credit (creditScore, creditHistory)

For the model classes, each model here includes important data the user should be able to save to the program. The specific goal is to create a computer application that stores the data regarding the user's employment status, the assets they own, the liabilities that they have, and their credit. These include the subclasses of loans and bills (these are from the superclass of liability). Each model object should be created by the user and saved in the program. If the program uses only one model class, that model class is the only one needed for this persistent data strategy.

Other classes needed:

- financeList (ArrayList<Object> fList; This is currently only defined as assetList)
- financeController (creates the functionality of all views)

In this persistent data strategy, a list is needed to store all of the model objects in the program. Currently, there is only a model object created to store Asset objects. If this program stays and only uses one model object, then the assetList class can stay. If not, this needs to be changed so that it stores all of the model objects.

The financeController class is very important because it will be used to store all of the data that is typed by the user. This can be in any of the three views that are contained in the program. The Java view classes are very important for displaying the information to the user, but they are not included in a persistent data strategy. They are also used for the user to create new model objects for the program. The test classes are also not necessary for this strategy. They just test to see if the object classes contained in the model package work. So six object classes need to be implemented in this strategy (depending on how many are utilized in the program), a list that stores each object created (or one), and the main controller to create persistent data..