# 4

Unfortunately, I wasn’t able to get your code running on my machine. Here is some feedback on your code design but your first priority should be insuring that your code works for phase 2.

**Strong:**

The Operators package is very well designed, it will be easy to add new types of users and bank employees. You might want to consider using interfaces if you plan on adding new types of employees, to ensure your code remains flexible.

**To improve your mark for phase 2:**

The Model.java class is both very long and has many long methods in it. Is there a better way of designing your menus? What are the flaws with the current way you’re doing it? Would it be easy to add more options or new types of users that have their own sets of options? How would you fix that? Fixing this class will be one of the keys for getting a good mark on phase 2.

Your transactions class involves handling multiple different types of transactions. Whenever you use it, you’ll only be using it to handle a single type of transaction (but you’ll have to create an instance of the entire class, much of which will not be used given whatever context you use it in). There seems to be a natural inheritance structure here that might lead to more organized and better performing code.

Consider using interfaces to ensure that certain types of account have appropriate behavior (a “transferable” interface for example). How would that make your code more flexible?

**Disclaimer:**

Please note that these suggestions do not constitute a full diagnosis of the design of your code. I did not have the time to fully dive into the details of every aspect of your code to be absolutely certain about what might make for good or bad design. These are merely suggestions/questions to help give you a better sense of how you should think when critiquing your own code. If you disagree with any of my assessments, feel free to talk to me about it in the tutorial.