Candidate Coding Challenge

CONTEXT

Landbay is a marketplace mortgage provider in the FinTech space that invests funder's money into mortgages.

Some context first:

- A mortgage is a loan from Landbay to a borrower. When borrowers apply for a mortgage, they select how much they want to borrow as well as the type of product.
- Products have different rates based on the risk that Landbay takes. For example a student let property may come at a higher risk (due to seasonality for example) than a semi-detached house in a highly residential area. As such, higher risk products will have higher rates.
- Landbay generally has around 100 products on sale at any point in time to try and cover as big-a-breadth of the market as possible. Landbay therefore sells a diverse set of products.
- Landbay does not lend its own money, and as such Landbay works with multiple funders (large banks or other types of financial institutions). Each funder will have a different appetite for risk and so Landbay allows the funder to select which products they want to fund. Some funders will therefore fund the same product.
- It's important to Landbay that the products it sells can be funded by at least one funder.
- Funders care more about deploying large amounts of money than they do about the profit/return they make on each mortgage. This is because cash sitting in a bank account doesn't earn any interest.
- Landbay's main objective is to fund as many mortgages as possible.

TASK AT HAND

In order to operate fairly, Landbay wants a system that does the following:

- Allocates mortgages to each of the funders, based on the product that was selected
- Ensures the allocation is as fair as possible between funders

We have provided some sample files of:

- products - the list of products that Landbay could sell
- funder mappings - a list of mappings that describes the products each funder is willing to fund
- mortgages - the list of mortgages that we want to fund

The purpose of the exercise therefore is to develop a small application that will output the best (or 'a' best) allocation of mortgages to funders. The app can be a simple command line based app and can output the results on the command line also.

In addition, describe what ***"fairness"*** algorithms you envisaged and why you chose the one you implemented.

BONUS

Can you describe how or adapt your app to do the following:

- Funders want to avoid having high concentration of mortgages in any specific location. This is because property market changes can overly affect some areas more than others and put their investments at risk.
- Landbay may prefer to fund mortgages through one funder rather than another. This is because Landbay may make more profit with particular funders. This is usually done in agreement with the funders but opens up the opportunity to have a product funded 60% by Funder A and 40% by Funder B, etc.

OTHER INFORMATION

The candidate should go as far as he can and use the exercise to demonstrate good OO principles, design principles and good programming practices. It is also an opportunity to demonstrate what good production-ready code looks like.

Making further assumptions and/or not implementing all the possible cases is perfectly acceptable, but this should be documented.

As a result of the exercise, we expect the following:

- The code is written in Java
- A README file that describes how to run the program
- A program that takes 3 (or more) input files and that prints out on the command line the results of the funder matching

- You are free to use any other libraries or technologies that you like, so long as we are able to run the system in a single command. For example you may want to use OpenCSV or SuperCSV to load the sample files provided.
- You free to use Maven, Gradle etc.
- You can do the entire exercise in memory and no data needs to be persisted in any database.