- Have a GitHub repository, with commits from all teammates.
- This repository should have a README.md describes your project. 10/10
- Create a class diagram in draw.io and include it in your README.md 10/10
- Start with the program that collects accounts.
- Initialize your accounts by reading the provided JSON file, accounts.json Download accounts.json. Allow the user to create additional accounts with the GUI. 10/10
- Do not allow the user to create duplicate accounts, or duplicate account numbers. 10/10
    - Add an account number attribute to class Account, and a corresponding field to the GUI.
    - Place all used account numbers in a Set.
    - When the user enters an account, validate that it is not currently in the Set of account numbers.
- Allow the user to withdraw money from accounts. Start by withdrawing from the account that pays the lowest interest
    - Adjust the GUI to allow withrdawls. 10/10
    - Place accounts in a priority queue, based on rate.
    - Use Comparator/compareTo to compare rates. 8/10
    - Allow partial withdraws from accounts; simply update principle.
- Report total interest earned, across all accounts, for a given period. 10/10
    - Iterate over the collection. Compute interest earned. Store interest earned from each account in a collection (ArrayList, etc.) Iterate and sum.
- Test your work. 10/10
    - Write unit tests to cover the cases above.
    - Ensure they run with GitHub actions.
- Code is in good form, and conforms to best practices. Classes and public methods contain JavaDoc. 8/10
- Do something extra, beyond the minimum requirements stated here. 10/10
    - What did you do? Must list to earn credit.

Fetch and update an existing account, you can also delete an account if you created it in error by selecting from the list and deleting with the specified button!