Create an object-oriented program (the inheritance, abstract class and abstract functions must be applied here) that allows user to manage (view, add, reset, update) the bank accounts information that he/she is the representative person.

There are two type of bank account that we need to manage: saving accounts and loan accounts.

* The saving accounts are to track the saving moneys
* The loan accounts are to track the borrowing moneys

For a saving account, there are some data constraints as below:

* The saving amount must be positive and greater than or equal to 100.000.000 VND
* The term must be positive and >= 30 days

For a common loan account, there are some data constraints as below:

* The amount borrowed must be positive and less than or equal to 1.000.000.000 VND
* The term must be positive and <=365 days
* Interest rate must be less than 15%

There are two special types of loan account that the system needs to handle, the personal loan and the investment loan:

* In the personal loan, the amount borrowed must be less than or equal to 100.000.000 VND, the term must be <= 30 days, and the interest rate must be between 5% and 15%
* In the investment loan, the amount borrowed must be <= 500.000.000 VND, the term must be less than or equal to 90 days, and the interest rate must be less than 18%

The application includes features as listed below. It would allow users to choose one of those features to handle when starting the application, when completing a feature, or when getting an error while in a feature progress. The user can also choose to exit the application.

* 1. View bank accounts: the system prints current user’s bank accounts (including account date, account type, and the account status)
  2. View account information: the system shows current user’s bank accounts, ask user to choose the bank account (by inputting account date, and the account type of loan or saving) and then shows the detailed information of the selected bank account
  3. Add bank account: the system allows users to input and store a new bank account (including account date, account type, amount, the term, the interest rate) into the system. The account date user inputted must be unique for each account type (loan, saving).
  4. Reset an account: the system shows current user’s bank accounts, ask user to choose the account (by inputting account date, and the account type of loan or saving), then reset the account amount to the new value (which is gained by adding the up-to-date interest to the current amount) and update the account date to the current date accordingly.
  5. Update account amount: the system shows current user’s bank accounts, ask user to choose an account to update (by inputting account date, and the account type of loan or saving) and enter the updated amount, then reset the selected account’s amount by subtracting/adding the entered amount from/to the up-to-date amount (which is gained by adding the interest amount, counted to current date, to the current amount), and update the account date to the current date. Besides, the system would de-activate (by change the account status to de-activated) the selected account in case the new account amount is 0.