Program design:

This program’s goals are to act as a investment bank, the client can open accounts, withdraw funds, deposit funds, transfer funds, or ask for the transactional history to be printed. The client is identified by their unique 4 digit id’s. A fifth digit can be added to pull up their unique investment account. And 5 types of transactions can be performed; Open, Deposit, Withdraw, Transfer and display history. The third phase, you would print out each client account and the amount held in each specific account. You are expected to store client information in binary trees.

JollyBanker.py

Variables: (all 10 funds, stored as floats), client\_account(int), transactions(Queue), client\_account\_balance(float), client\_name(str), account\_list(Binary tree)

Methods: transaction(operation(char), account\_number(int), money\_amount(int)), transaction(operation(char), name(str), account\_number(int)), transaction(operation(char), account\_number(int)), transaction(operation(char), money\_amount(int)), transaction(operation, origin\_account(int), money\_amount(int), destination\_account(int)), open(name(str), id(int)), Deposit(account\_number(int), money\_amount(int)), withdraw(account\_number(int), money\_amount(int)), transfer(origin\_account(int), money\_amount(int), destination\_account(int)), history(account\_number(int))