SavingsAccount

CheckingAccount

BankAccount

Here is the class hierarchy for

bank accounts from Assignment 7B:

Make sure your assignment 7B is working, copy it to a new 7D project, and make changes as follows:

1. Add the capability for all bank accounts in the hierarchy to have a **name** (a String) and an **account number** (also a String). They should get passed as parameters in and set upon construction. Write methods that allow accounts to have their name and account number retrieved (these kinds of methods usually start with **get**). Adhere to good object-oriented principles when deciding which class(es) to change. Answering the following question might help: Should this new functionality be handled in the parent class and inherited to the child classes, or should it be handled in the child classes? If you're not sure, clear it with me.
2. Add the capability for all bank accounts to return a human readable string (**toString**). The human readable string should ONLY have *name*, *account number*, and *balance* as shown. Follow good object-oriented principles when deciding which class(es) to change for this (should it be in the parent or in the children?)

Michael 16-18236 $3,870.01

Note, a good way to get the formatting to look like this (above) is to use (I’ll explain this if you ask):

return String.format("%-10s %7s $%,7.2f", name, acctNum, balance);

1. I have included an AccountClient and a file called accts.txt; import them to your project. AccountClient opens a the disk file 'accts.t.xt', reads it, creates a bunch of SavingsAccount /CheckingAccount objects, and puts them all into an ArrayList.

It will only work if your account classes adhere to these things:

* 1. The names of your classes match exactly the diagram above.
  2. The order of parameters for the CheckingAccount constructor is (1) balance, (2) name, (3) acct number.
  3. The order of parameters for the SavingsAccount constructor is (1) balance, (2) interest, (3) name, (4) acct num.

**BIG IDEA:** The ArrayList is declared to hold parent BankAccount object references, but the actual objects referenced are a mix of child objects (CheckingAccount and SavingsAccount objects). This is where the power of inheritance and polymorphism comes from: we are processing a list of parent types, but the actual objects referenced will take care of themselves the way they know how to. We don’t have to worry about it in the client!

Follow the numbered comments in the client to complete it.

1. Try to comment out the abstract **endOfMonth** method in the parent BankAccount. This will create an error in the client, because it thinks that parent BankAccount references will not have an endOfMethod method. We need that declared in the parent so our client will be happy, and so the polymorphism can take place. Go back and take the comment out….