Java 3 Portfolio Activity 1

30003389 – Kyer Potts

# Advantages of ArrayList over traditional structures

* ArrayLists are resizable arrays, traditional arrays have all memory resources allocated at initialisation, which prevents additional array elements from being added or the array size from being expanded upon.
* Elements in an array list can be inserted into index positions declared by the programmer. This allows for dynamic organisation of the ArrayList. Traditional array elements are added at the end of the array, and require separate functions to re-organise the elements within the array.
* Unless specifically declared with generic types, ArrayLists can contain any type of object simultaneously. Arrays can only contain their declared type.
* ArrayLists have class methods that perform specific functions that are extremely helpful such as organisational, search and modification functions.
* Elements in ArrayLists can be traversed in both directions using ListIterator objects

# What Happens When You:

### Display Accounts:

If no accounts have been created, the program Displays “No Accounts currently created” in the Account Details field. If Accounts have been created, it displays the Account Name, Account Number, and the Balance in the Account Details field for all accounts that have been created

### Create Accounts:

If the Name field and Account Number field are empty, a message advising that user to fill out the required fields shows in the Account Details output control. If those fields are completed and the Create button is clicked, a new Account containing the information from the required fields is created and then shown in the Account Details output control.

### Delete Accounts:

A message is displayed in the Account Details output control advising that the delete function is not coded in this version.

### Make a Withdrawal

If the Account Number and Withdraw fields are filled out with information, the “Make Transaction” button will remove the amount specified from the withdraw section from the account containing the specified account number.

### Make a Deposit

If the Account Number and Deposit fields are filled out with information, the “Make Transaction” button will add the amount specified from the deposit section to the account containing the specified account number.

# Can you display accounts before they are created? Why or why not?

The display accounts button will not cause any errors to occur within the program itself, however it will provide a message to the user that no accounts have been created, and no information pertaining to any account will be shown in the Account Details output control. This is coded into the application as an error handling measure intended to prevent crashes.

# Can you create an account before entering anything in the fields? Why or why not?

An account cannot be created without first entering information into the required fields (Name and Account Number). These are required fields for accounts and must be completed in order to generate an account this is coded into the program and will generate a message to the user explaining that these fields are required in order to create an account. The program provides a default balance of 0, however can be overridden with an amount entered into the balance field.

# Can you make a withdrawal transaction with no amount in the withdrawal field? Why or why not?

The program will not crash or generate any error messages, however it will also not perform any function as the program has not been provided an amount to deduct from the accounts current balance. The program decides whether a deposit or transaction has been made by identifying whether the deposit field is greater than 0 or whether the withdraw field is greater than 0. If neither of these fields are populated with information, the program will not be able to perform the function correctly.

# Can you make a deposit transaction with no amount in the deposit field? Why or why not?

Once again the program will not crash, as the code dictates that for the function to perform, an amount greater than 0 has to be entered into either into the Withdraw or Deposit fields. If no amount has been entered the amount is considered null. This will not crash the program, as the program logic assumes that null is equal to or less than 0, however no amount will be removed or added to the current balance of the account.

# What additions would you make to the current application to increase it’s functionality?

Currently the program will crash if you enter an amount into the Deposit and Withdraw fields simultaneously and then click the make transaction button. A simple way to prevent this is to add basic error handling if the Deposit and Withdraw fields both have amounts that aren’t “0” or “null”. Another way to prevent this error for occurring is to divide the transaction button into it’s two respective types, one button for withdraw, which performs any withdraw function and only subtracts desired amounts from the account balance, and another button for deposit, which will only add desired amounts to the account balance.

The Delete button could also been made to work by checking to see whether an account number has been added into the account number field, and whether or not that particular account still has a balance. If the account exists and the current balance is 0, then it will remove the account from the account array by running a binary search to return the index of the account and then deleting that index from the array and shifting the rest of the elements up by 1 index to close the “null” gap left by the deleted account element.