Binary Search Tree Lab
(Bank Account)
Advanced Computer Science

Topics: Objects, Classes, Java Components, Binary Search Tree Linked

Program Description

You will create a bank account administrator and customer program that will keep track of user accounts. You will use a **Binary Search Tree** to keep track of user information. You cannot import and use any other data structure.

Account Class implements Comparable

This class will contain information about the user. The information will be the last name, first name, pin, and account balance. This class will implement the Comparable interface. The Comparable interface will be used to compare the last name and then the first name alphabetically from a to z.

Node<E> class

This node will be compatible with a binary search tree.

BinaryTree<E Extends Comparable<E>> Class

This class will have the structure and functionalities of a Binary Search Tree for search, insertion, and deletion. Nodes are sorted using the compareTo method of the Comparable Interface. There cannot be two nodes with both the same last and first name.

Screen Class

This class will be used to display the program.

Runner Class

Runs the program

Add more classes as needed.
Add more methods and instance variables as needed for each class.

Features:

Create buttons, tabs, or use any approach you want to switch between the admin and customer view.

Admin View

1. Your program will have a scroll bar pane listing all the user bank accounts showing only the last and first name. You will add accounts from this list of last name, first name (link). For the account balance, give them a random number from 0 to 100,000 as a double. For the pin, use random numbers from 0000 to 9999. **This list will be sorted from a to z by the last name and then the first name.**
2. There will be inputs for the user to add a new account. The user will input the last name, first name, pin, and balance. **Display the number of passes** it took to add any given account. Each method call will count as a pass.
3. There will be inputs for the user to search for an account given the last and first name. When searching for an account, you will clearly **display the number of passes** it took to find the account. Each method call will count as one pass.
4. Once an account is found, display the full name, balance, and pin. There will be options to delete the account, change the first name, last name, pin, and the balance. When there is a change, the tree will be updated accordingly.
5. Information will automatically save to file when there is a change. When you run the program again, it will have all the latest information. How you save to file is up to you. You can save the object to file in which you will need to implement serializable, or you can write the data to a text file.

Customer View

1. Have inputs for the customer to enter in their first name, last name, and pin. You will search for the account, and check the pin given by the user. **Display the number of passes** it took to find the account.
2. Once a login is successful, the name and balance will be displayed. Then, there will be options to change the first name, last name, change the pin, deposit, and withdraw. Any changes will be reflected on the screen. When there is a change, the tree will be updated accordingly.
3. Information will automatically save to file when there is a change. When you run the program again, it will have all the latest information. How you save to file is up to you. You can save the object to file in which you will need to implement serializable, or you can write the data to a text file.
4. Have a logout button where it takes the user back to the menu where another customer can log.

Grading Rubric:

Your lab will be graded using the following criteria.

Completion of Lab as Described	90

~~Milestone 1 **- Due Friday In the Admin View,** You can enter in the last and the first name, and the corresponding user account information is displayed. In addition, display the number of passes it took to locate the account.~~	10

You will have class time to work on this lab. Any late submission or revision will be given a maximum score of 90.

Due Date and Submission: See Google Classroom. Link