Various Operations Of Banking Account Using Data Structure

LJ University
University with a Difference

❖ In this project, we have designed and implemented a Java program to simulate banking operations, showcasing the importance of efficient data structures in the world of finance.

Created By :-

208	Kewalramani Dev
184	Om Changani
192	Dhruvil Parmar
206	Gohel Hardik
207	Dhvanit Vaghela

Main Objectives of the Project

- Efficient data structures enable banks to manage vast amounts of customer data, transactions, and accounts effectively and are crucial for ensuring fast and accurate retrieval of customer information, transactions, and account balances.

1. **Create Customer Accounts:** Implement the ability to create new customer accounts, each with a unique account number, name, and initial balance.
2. **Perform Transactions:** Enable customers to deposit, withdraw, and transfer funds between accounts while maintaining data accuracy and security.
3. **Track Transaction History:** Develop a transaction history feature to record and view all transactions performed within the system.
4. **Delete Customer Accounts:** Implement the capability to delete customer accounts while maintaining the integrity of the data.
5. **View Customers in Dictionary Order:** Allow users to view all customer details in dictionary order based on customer names.

Data Structures Used In Project

Singly Linked List:

- Role: Used to maintain a list of customer accounts.
- Why Chosen: Chosen for its simplicity and efficiency in dynamic memory allocation. It allows for easy addition of new accounts and sequential traversal.

Circular Linked List:

- Role: Used to implement a circular customer list for efficient insertion and retrieval of customers.
- Why Chosen: Circular lists provide a continuous loop of customers, allowing quick access to both the front and rear elements. Ideal for managing customer queues.

Stack:

- Role: Used for recording and managing transaction history.
- Why Chosen: A stack is a natural choice for tracking transactions since it follows the Last-In-First-Out (LIFO) principle, making it suitable for undoing the most recent transaction.

Data Structures Used In Project

Queue (Circular Queue):

- Role: Used as a customer queue for processing customer requests in a first-come, first-served manner.
- Why Chosen: Circular queues efficiently manage customer arrivals and departures, providing a fair and predictable order of service.

Binary Search Tree:

- Role: Used for efficient searching and retrieval of customer accounts based on their account numbers.
- Why Chosen: Binary search trees offer logarithmic time complexity for search operations, ensuring fast and efficient account retrieval based on account numbers.

- Data Structure Selection: Choosing the right data structures for specific functionalities was challenging. We needed to consider efficiency, complexity, and practicality.

Potential Improvements for the Project:

Implementing a User Interface (GUI):

- Enhancement: Develop a graphical user interface (GUI) for the banking system to provide a user-friendly and intuitive interface for customers and bank employees.
- Benefits:
  - Improved user experience, making it easier for customers to interact with the system.

Enhancing Error Handling and Validation:

- Enhancement: Strengthen error handling and validation mechanisms to detect and prevent common input errors, data inconsistencies, and system failures.
- Benefits:
  - Improved data integrity and accuracy.
  - Reduced risk of erroneous transactions and potential financial losses.

Thanks!