```
/**
 *  The BankOperation classs has the following fields
 *  - theBank of Bank type
 *  - 2 objects of Clerk type
 *  - The initialBalance array is of int type. It denotes the amount in the two accounts. It has 2
values - 500 and 800 .
 *  - Initialize totalCredits integer array with initialBalance array length.
 *  - Initialize totalDebits integer array with initialBalance array length.
 *  - Initialize transactionCount of int type with  value 20 .
 *  - Declare accounts as an array of Account type.
 *
 *
 */

/**
         * The Constructor BankOperation() initialises theBank object.
         * It initialises the clerk object (both of them) and takes bank object as an argument.
         * Initialize the accounts array with initialBalance array length.
         *
*/

/**
 * Run a for loop to create accounts by initialising the account array elements with the Account
Constructor of Account class.
 * Pass paramters i+1 and  initialBalance[i] in the calling constructor.
 * Initialise totalCredits[i] and totalDebits[i] with 0.
 */

/**
 *   createThread() - This method creates 2 threads clerk1Thread and clerk2Thread and pass the
respective clerk objects.
 *
 */

/**
 * Make the clerk1Thread to run as background thread by calling the setDaemon method.
 */

/**
 * Make the clerk2Thread to run as background thread by calling the setDaemon method.
 */

/**
 * Start the first clerk1Thread thread.
 */

/**
 *Start the first clerk2Thread thread.
 */

/**
 * createTransaction() method creates a reference variable of Transaction class.
 * Declare two variables select and amount and initialise them with 0 and 50 respectively.
 *
 */

/**
 * Run a for loop from 1 to transactionCount to initialise transaction object with the constrcutor of
Transaction class.
 * Pass parameters accounts[select], Transaction.CREDIT, amount in the transaction constructor.
 * Keep total credit record by adding amount to the totalCredits[select].
 */

/**
```

```
   * run a while loop to wait until the first clerk (clerk1) is busy.
   * If the clerk is busy the thread sleeps for 25 units.
   * Handle the InterruptedException.
   */

  /**
   * Call the doTransaction() method on clerk1 object. Pass transaction as parameter to this object.
   */

  /**
   * Initialise amount =30 here.
   * Initialise transaction object with the constrcutor of Transaction class.
   * Pass parameters accounts[select], Transaction.DEBIT, amount in the transaction constructor.
   * Keep total debit record by adding amount to the totalDebits[select].
   */

  /**
   * Run a while loop to wait until the second clerk (clerk2) is busy.
   * If the clerk is busy the thread sleeps for 25 units.
   * Handle the InterruptedException.
   */

  /**
   * Call the doTransaction() method on clerk2 object. Pass transaction as parameter to this object.
   */

  /**
   * If either of the two clerks is busy, run a while loop and let the thread sleep for 25 ms of time.
   * Handle the InterruptedException
   */


  /**
   * getaccountnumber() method returns array of the account number by calling getAccountNumber()
  method.
   * @return
   */

  /**
   * getcredit() method returns totalCredits.
   * @return
   */

  /**
   * getdebit() method return totalDebits.
   * @return
   */

  /**
   * getaccounts() method returns array of the account balance by calling getBalance() method.
   * @return
   */
```