HW8: Bank Accounts: Exceptions and Advanced File I/O

Modify HW7 by changing how the array of Accounts is kept.

The Classes needed are:

1. A **Bank** class which consists of an **array of Account records (kept in a binary file - see below)** and the **number of accounts** currently active in the bank. It also contains the static variables:
   totalAmountInSavingsAccts - sum total of balances in all Savings accounts
   totalAmountInCheckingAccts - sum total of balances in all Checking accounts
   totalAmountInCDAccts - total - sum total of balances in all CD accounts
   totalAmountInAllAccts - total - sum total of balances in all accounts
2. An **Account** class which consists of a **Depositor, an account number, an account type, account status (open or closed), account balance, and an array of TransactionReceipts.**
3. A **Depositor** class which has a **Name and a social security number**.
4. A **Name** class which consists of **first and last names**.
5. A **Check** class with data fields consisting of **an account number, the check amount, and a dateOfCheck**
6. A **TransactionTicket** class with data fields consisting of **a dateOfTransaction, typeOfTransaction (deposit, withdrawal, balance inquiry, new account, delete account, etc.), amountOfTransaction (for deposits and withdrawals), termOfCD (6, 12, 18, or 24 months - see below)**.
7. A **TransactionReceipt** class with data fields consisting of **a TransactionTicket, successIndicatorFlag, reasonForFailure String, preTransactionBalance, postTransactionBalance, postTransactionMaturityDate (for CDs).**

The array of Accounts is to be kept as a sequence of fixed length Account records in a Random Access Binary File named BankAccounts.dat

The array of TransactionReceipts for each account is to be kept if a separate binary file. Each TransactionReceipt is to consist of a fixed length record. Each file is to consist of a sequence of TransactionReceipts for a particular account.

When a new account is opened, a new Account record is appended at the end of BankAccounts.dat file and the number of accounts currently active is incremented.

When an account is to be deleted, the deleted Account record in the file is to be replaced by the last active Account record of the file and the number of accounts currently active is decremented. The file is to be truncated using the RandomAccessFile method setLength(long newLength).

Invalid operations are to be handled by exceptions. You should minimally implement the following exceptions:
InvalidAccountException
InvalidAmountException
AccountClosedException
InsufficientFundsException
InvalidMenuSelectionException
PostDatedCheckException
CheckTooOldException
CDMaturityDateException

As in previous assignments, make sure to use enough test cases so as to completely test program functionality.

Submission Requirements:
Create a folder on Google Drive that will contain the following:
1. The source files (i.e., *.java files) for each of the implemented Classes:
pgmHW8.java
Bank.java; Account.java; Depositor.java, Name.java
Check.java; TransactionTicket.java; TransactionReceipt.java;
SavingsAccount.java; CheckingAccount.java; CDAccount.java;
implemented Exception Classes
etc.
2. The text file containing the initial database of accounts (e.g., initAccounts.txt)
3. The test cases text file (e.g., myTestCases.txt)
4. The output text file which contains all of the required program output (e.g., pgmOutput.txt)
5. The BankAccounts.dat file and all of the TransactionReceipts binary files
Then, make the folder shareable and send me a link to the folder.