INSTRUCTION:

Design and implement a Java application for the application described below. You must apply `State` Design Pattern.

Consider a business account at a bank with the overdraft facility. Such an account can exist in any one of the following three states at any given point of time:

No trans fee

1. *No transaction fee state* — As long as the account balance remains greater than the minimum balance, no transaction fee will be charged for any deposit or withdrawal transaction. The example application has the minimum balance as $2,000.

trans fee

2. *Transaction fee state* — An account is considered to be in the transaction fee state when the account balance is positive but below the minimum balance. A transaction fee will be charged for any deposit or withdrawal transaction in this state. The example application has the transaction fee in this state as $2.
3. *Overdrawn state* — This is the state of the account when an account balance is negative but within the overdraft limit. A transaction fee will be charged for any deposit or withdrawal transactions in this state. The example application has the transaction fee in this state as $5 and the overdraft limit is maintained as $1,000. A message "Attention: Your Account is Overdrawn" will be displayed if the account is overdrawn.

Overdrawn

In all three states, a withdrawal transaction that exceeds the overdraft limit is not allowed. Figure 1 depicts possible state transitions for an account and Table 1 shows how these transitions can occur.

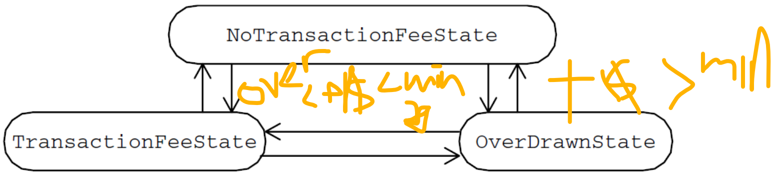
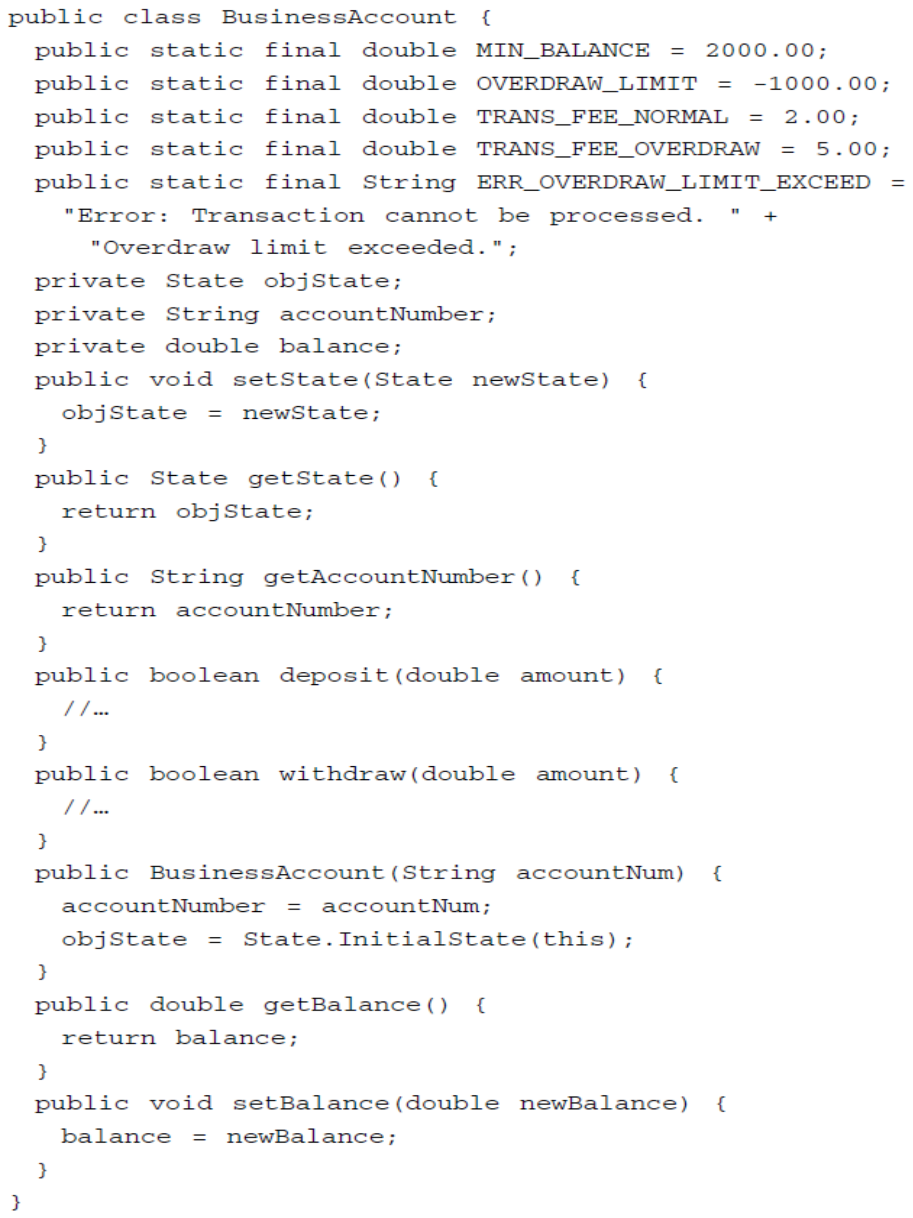


Figure 1: State Transitions among Different Account States

Table 1: State Transitions among Different Account States

From	*To*	*What Causes the Transition*
No transaction fee state	Transaction fee state	A withdrawal that can make the balance positive but less than the minimum balance.
	Overdrawn state	A withdrawal that can make the balance negative.
Transaction fee state	No transaction fee state	A deposit that can make the balance greater than the minimum balance.
	Overdrawn state	A withdrawal that can make the balance negative.
Overdrawn state	No transaction fee state	A deposit that can make the balance greater than the minimum balance.
	Transaction fee state	A deposit that can make the balance positive but less than the minimum balance.

The `BusinessAccount` class offers the basic functionality in the form of methods to enable a client object to perform deposit and withdrawal operations. In addition, the `BusinessAccount` class defines some of the transaction limits and offers accessor methods to read its state. The `BusinessAccount` class is given below. You need to write the statements for its `deposit` and `withdraw` methods:



```
public class BusinessAccount {
  public static final double MIN_BALANCE = 2000.00;
  public static final double OVERDRAW_LIMIT = -1000.00;
  public static final double TRANS_FEE_NORMAL = 2.00;
  public static final double TRANS_FEE_OVERDRAW = 5.00;
  public static final String ERR_OVERDRAW_LIMIT_EXCEED =
    "Error: Transaction cannot be processed. " +
      "Overdraw limit exceeded.";
  private State objState;
  private String accountNumber;
  private double balance;
  public void setState(State newState) {
    objState = newState;
  }
  public State getState() {
    return objState;
  }
  public String getAccountNumber() {
    return accountNumber;
  }
  public boolean deposit(double amount) {
    //…
  }
  public boolean withdraw(double amount) {
    //…
  }
  public BusinessAccount(String accountNum) {
    accountNumber = accountNum;
    objState = State.InitialState(this);
  }
  public double getBalance() {
    return balance;
  }
  public void setBalance(double newBalance) {
    balance = newBalance;
  }
}
```

1. Define a common `State` class that contains the business account behavior that is common across all states. Instead of keeping the state-specific behavior inside the `BusinessAccount` class, by applying the State pattern, the behavior specific to each of the three states can be implemented in the form of three separate subclasses - `NoTransactionFeeState`, `TransactionFeeState` and `OverDrawnState` of the `State` class. In addition, these subclasses know the state it should transition to and when to make that transition. Each of these `State` subclasses implements this state transition functionality by overriding the parent class `transitionState` method as per the state transition rules detailed in Table 1. `transitionState` method returns the `State` object that represents the current state of the `BusinessAccount` object after a transition of state.

   Apply `State` Design Pattern to design your system. Draw a UML class diagram to show your design for the system.

2. Implement your design in Java. Also implement a test client `AccountManager` to allow a user to perform different transactions on a business account. When executed, the `AccountManager` creates a `BusinessAccount` object that represents a business account and displays the necessary user interface to allow a user to perform deposit and withdrawal transactions that can make the business account go through different states. A sample output is given below.

```
Enter 1 to display account, 2 to deposit, 3 to withdraw, 0 to exit: 1
Account number = 1111 2222 3333 4444, Balance = 0.0
Enter 1 to display account, 2 to deposit, 3 to withdraw, 0 to exit: 2
Enter amount to deposit: 3000
An amount 3000.0 is deposited
Account number = 1111 2222 3333 4444, Balance = 3000.0
Enter 1 to display account, 2 to deposit, 3 to withdraw, 0 to exit: 2
Enter amount to deposit: 2000
An amount 2000.0 is deposited
Account number = 1111 2222 3333 4444, Balance = 5000.0
Enter 1 to display account, 2 to deposit, 3 to withdraw, 0 to exit: 3
Enter amount to withdraw: 4000
An amount 4000.0 is withdrawn
Account number = 1111 2222 3333 4444, Balance = 1000.0
Enter 1 to display account, 2 to deposit, 3 to withdraw, 0 to exit: 3
Enter amount to withdraw: 500
Transaction Fee ($2.0)was charged due to account status (less than minimum balance)
An amount 500.0 is withdrawn
Account number = 1111 2222 3333 4444, Balance = 498.0
Enter 1 to display account, 2 to deposit, 3 to withdraw, 0 to exit: 3
Enter amount to withdraw: 1000
Transaction Fee ($2.0)was charged due to account status (less than minimum balance)
Attention: Your Account is Overdrawn
An amount 1000.0 is withdrawn
Account number = 1111 2222 3333 4444, Balance = -504.0
Enter 1 to display account, 2 to deposit, 3 to withdraw, 0 to exit: 3
Enter amount to withdraw: 200
Transaction Fee ($5.0)was charged due to account status(Overdrawn)
An amount 200.0 is withdrawn
Account number = 1111 2222 3333 4444, Balance = -709.0
Enter 1 to display account, 2 to deposit, 3 to withdraw, 0 to exit: 3
Enter amount to withdraw: 800
Error: Transaction cannot be processed. Overdraw limit exceeded.
Account number = 1111 2222 3333 4444, Balance = -709.0
Enter 1 to display account, 2 to deposit, 3 to withdraw, 0 to exit: 5
Invalid choice. Please try again.
Enter 1 to display account, 2 to deposit, 3 to withdraw, 0 to exit: 0
```