CS2002-1

Lab Programs by

Harshith
NNM24IS092

Submitted to: Dr. Martis

Lab Program:

Scenario

In your college notice board system, one thread produces messages (like announcements), and another thread consumes them for display. The producer and consumer must coordinate using wait/notify.

Problem Statement

1. **Create a class MessageBoard with:**
   - **A private String message.**
   - **A boolean field hasMessage.**
   - **Method put(String msg) that waits if hasMessage is true, stores the message, sets hasMessage = true, and calls notify().**
   - **Method get() that waits if hasMessage is false, retrieves the message, sets hasMessage = false, and calls notify().**
2. **Create a Producer thread that sends 3 messages: "Exam on Monday", "Holiday on Tuesday", "Workshop on Wednesday".**
3. **Create a Consumer thread that prints each received message.**
4. **In the Main class, run both threads together.**

Github Link:https://github.com/Harshith161/Java-Progs

Code:

```
package bankapp;


import java.util.HashMap;


class Bank
{
  private HashMap<Integer, Double> accounts = new HashMap<>();


  public void createAccount(int accNo, double balance)
  {
```

```
    accounts.put(accNo, balance);
    System.out.println("Creating account "+ accNo +" with " + balance);
  }

  public void deposit(int accNo, double amt) {
    if (accounts.containsKey(accNo)) {
      double newBalance = accounts.get(accNo) + amt;
      accounts.put(accNo, newBalance);
      System.out.println("Deposit" + amt + "in account" + accNo + "Balance = " + newBalance);
    }
    else
    {
      System.out.println("Account" + accNo + "not found");
    }
  }

  public void withdraw(int accNo, double amt)
   {
    if (accounts.containsKey(accNo))
    {
      double balance = accounts.get(accNo);
   if (balance >= amt)
   {
     double newBalance = balance - amt;
      accounts.put(accNo, newBalance);
        System.out.println("Withdraw " + amt + " from account " + accNo + " Balance = " +
newBalance);
      } else {
        System.out.println("Insufficient balance in account " + accNo);
```

```
        }
    } else {
        System.out.println("Account " + accNo + " not found!");
    }
  }

  public void checkBalance(int accNo) {
    if (accounts.containsKey(accNo)) {
   System.out.println("Account " + accNo + ": " + accounts.get(accNo));
  } else
  {
        System.out.println("Account " + accNo + " not found!");
    }
   }
}

public class Main {
  public static void main(String[] args) {
    Bank bank = new Bank();

    bank.createAccount(1001, 5000);
    bank.createAccount(1002, 2000);

    bank.deposit(1001, 1000);
    bank.withdraw(1002, 500);

    System.out.println("Final balances:");
    bank.checkBalance(1001);
    bank.checkBalance(1002);
```

```
}
}
```

Output:

```
Creating account 1001 with 5000.0
Creating account 1002 with 2000.0
Deposit1500.0in account1001Balance = 6500.0
Withdraw 700.0 from account 1002 Balance = 1300.0
Final balances:
Account 1001: 6500.0
Account 1002: 1300.0
```