

Question

Time left 0:30:49

Question 1

Not yet
answeredMarked out of
1.00

Flag question

Design a java application to implement the banking process. Consider the shared variable NETBALANCE has the complete branch balance. Design the cashier1, cashier2, cashier3 threads to access this variable and allow the operations of credit and debit. Design a manager, senior manager and ATM machine threads, which also has the access to the shared variable and do the operations of credit and debit. Ensure the shared variable is used safely and no multiple access by various threads at time.

Code screen shot

```
1 package lab.assignments;
2
3 class bank {
4     int Netbalance=1000; // Currently the bank has the balance amount=10,000
5
6     // debitting the amount from the bank ( balance money reduces)
7     synchronized public void debit(String person_name,int amount) {
8         this.Netbalance = this.Netbalance-amount;
9         System.out.println(person_name + " is withdrawing Rs : "+ amount + " from the bank"
10            + " and the balance is : " + Netbalance);
11     }
12
13
14     // crediting the amount into the bank (balance money increases)
15     synchronized public void credit(String person_name,int amount) {
16         this.Netbalance = this.Netbalance + amount;
17         System.out.println(person_name + " is crediting Rs : "+ amount + " into the bank"
18            + " and the balance is : " + Netbalance + "\n");
19     }
20 }
21
```

```
Lab Assignments - Apache NetBeans IDE 12.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)

fat_exam.java
Source History
22 class banking_operations extends Thread {
23     String thread_name;
24     int amount;
25     bank obj_bank; // main class
26
27     banking_operations(String name_from_main,int amount_main, bank obj_main) {
28         this.thread_name = name_from_main;
29         this.obj_bank = obj_main;
30         this.amount = amount_main;
31     }
32
33     public void operations() { // calling the credit() and debit()
34         this.obj_bank.debit(this.thread_name,this.amount);
35         this.obj_bank.credit(this.thread_name,this.amount);
36     }
37
38     @Override
39     // to do both the process credit and debit operations by the cashier and the executives
40     public void run() {
41         this.operations();
42     }

```

```
Lab Assignments - Apache NetBeans IDE 12.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)

fat_exam.java
Source History
46 public class fat_exam {
47     public static void main(String[] args) throws Exception{
48         System.out.println("The current balance in the bank : 1000\n");
49         bank obj_bank = new bank();
50         banking_operations cashier_1 = new banking_operations("Cashier_1",10,obj_bank);
51         banking_operations cashier_2 = new banking_operations("Cashier_2",20,obj_bank);
52         banking_operations cashier_3 = new banking_operations("Cashier_3",30,obj_bank);
53
54         banking_operations manager = new banking_operations("Manager",40,obj_bank);
55         banking_operations senior_manager = new banking_operations("Senior Manager",50,obj_bank);
56         banking_operations atm_machine = new banking_operations("ATM_MACHINE",60,obj_bank);
57
58         // starting the thread
59         cashier_1.start();
60         cashier_2.start();
61         cashier_3.start();
62
63         try{ Thread.sleep(200);
64             manager.start();
65             senior_manager.start();
66             atm_machine.start();
67
68         }catch(Exception e){};
69     }

```

Output

```
Lab Assignments - Apache NetBeans IDE 12.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Output - Lab Assignments (run) x
run:
The current balance in the bank : 1000

Cashier_1 is withdrawing Rs : 10 from the bank and the balance is : 990
Cashier_1 is crediting Rs : 10 into the bank and the balance is : 1000

Cashier_3 is withdrawing Rs : 30 from the bank and the balance is : 970
Cashier_3 is crediting Rs : 30 into the bank and the balance is : 1000

Cashier_2 is withdrawing Rs : 20 from the bank and the balance is : 980
Cashier_2 is crediting Rs : 20 into the bank and the balance is : 1000

Manager is withdrawing Rs : 40 from the bank and the balance is : 960
Manager is crediting Rs : 40 into the bank and the balance is : 1000

ATM_MACHINE is withdrawing Rs : 60 from the bank and the balance is : 940
ATM_MACHINE is crediting Rs : 60 into the bank and the balance is : 1000

Senior Manager is withdrawing Rs : 50 from the bank and the balance is : 950
Senior Manager is crediting Rs : 50 into the bank and the balance is : 1000

BUILD SUCCESSFUL (total time: 0 seconds)
```

Actual Code

```
package lab.assignments;

class bank {

    int Netbalance=1000; // Currently the bank has the balance amount=10,000

    // debitting the amount from the bank ( balance money reduces)
    synchronized public void debit(String person_name,int amount) {

        this.Netbalance = this.Netbalance-amount;

        System.out.println(person_name + " is withdrawing Rs : "+ amount + " from the bank"
            + " and the balance is : " + Netbalance);

    }

    // creditting the amount into the bank (balance money increases)
    synchronized public void credit(String person_name,int amount) {

        this.Netbalance = this.Netbalance + amount;

        System.out.println(person_name + " is crediting Rs : "+ amount + " into the bank"
            + " and the balance is : " + Netbalance + "\n");

    }

}

class banking_operations extends Thread {

    String thread_name;

    int amount;

    bank obj_bank; // main class

    banking_operations(String name_from_main,int amount_main, bank obj_main) {

        this.thread_name = name_from_main;

        this.obj_bank = obj_main;

        this.amount = amount_main;
```

```

    }

    public void operations() { // calling the credit() and debit()
        this.obj_bank.debit(this.thread_name,this.amount);
        this.obj_bank.credit(this.thread_name,this.amount);
    }

    @Override
    // to do both the process credit and debit operations by the cashier and the executives
    public void run() {
        this.operations();
    }
}

```

```

public class fat_exam {

    public static void main(String[] args) throws Exception{

        System.out.println("The current balance in the bank : 1000\n");

        bank obj_bank = new bank();

        banking_operations cashier_1 = new banking_operations("Cashier_1",10,obj_bank);
        banking_operations cashier_2 = new banking_operations("Cashier_2",20,obj_bank);
        banking_operations cashier_3 = new banking_operations("Cashier_3",30,obj_bank);

        banking_operations manager = new banking_operations("Manager",40,obj_bank);
        banking_operations senior_manager = new banking_operations("Senior
Manager",50,obj_bank);

        banking_operations atm_machine = new banking_operations("ATM_MACHINE",60,obj_bank);

        // starting the thread
        cashier_1.start();
        cashier_2.start();
    }
}

```

```
cashier_3.start();
```

```
try{ Thread.sleep(200);
```

```
    manager.start();
```

```
    senior_manager.start();
```

```
    atm_machine.start();
```

```
}catch(Exception e){};
```

```
}
```

```
}
```