Aima khan
2023F-BSE-64

LAB 5
INTER-THREAD COMMUNICATION

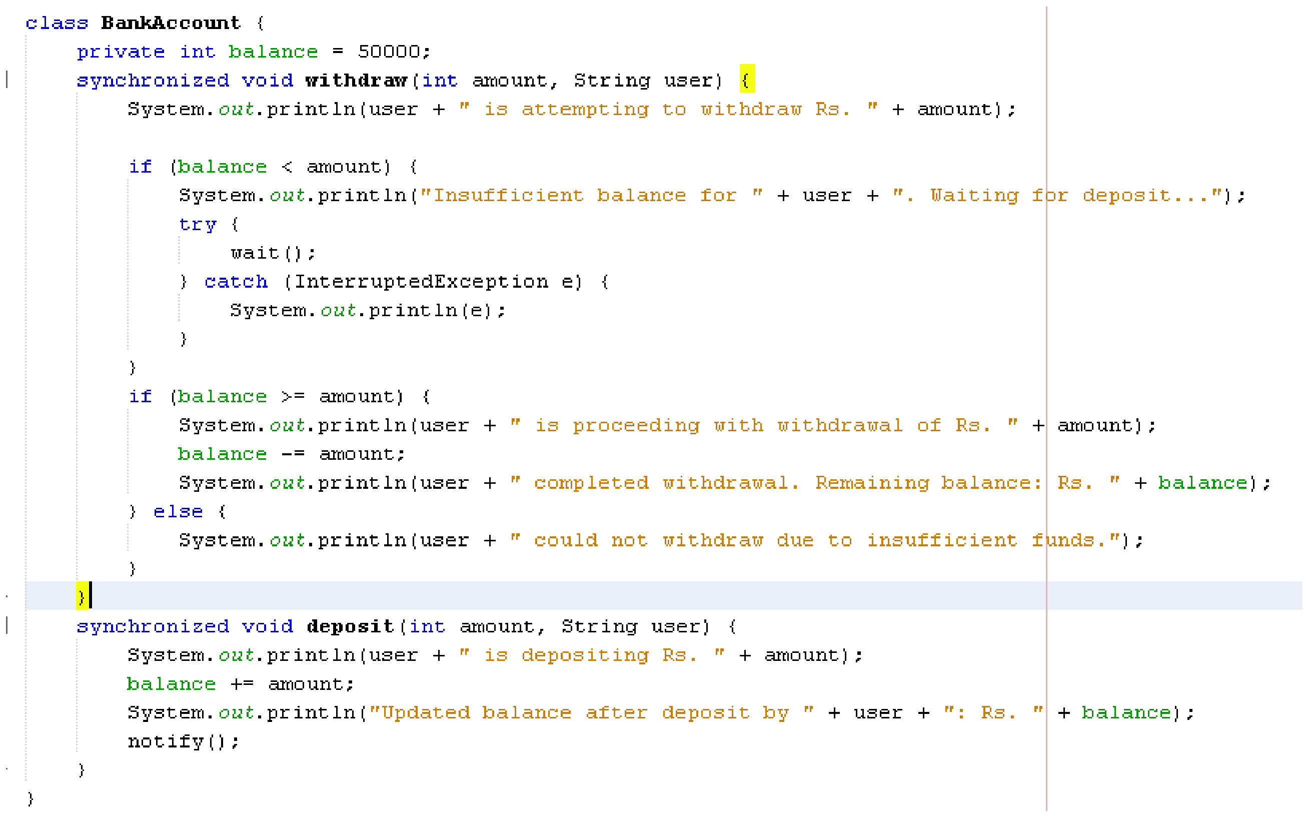
OBJECTIVE:

Develop an inter thread user program by synchronization

Lab Task:

1. Design a simple program of concurrency by implementing the scenario of two account holders in a joint bank account. (Hint: Total amount will be 50000, if 'user A' wants to withdraw 45,000 and 'user B' wants to withdraw 20,000) Apply mechanism of synchronization e.g. Block or Method for handling accessibility of multi-threads:

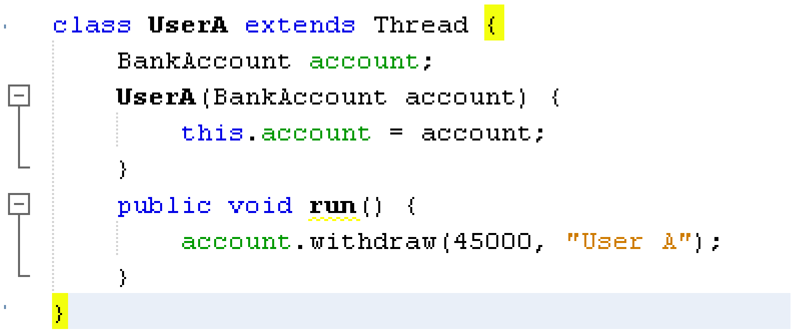
CODE:



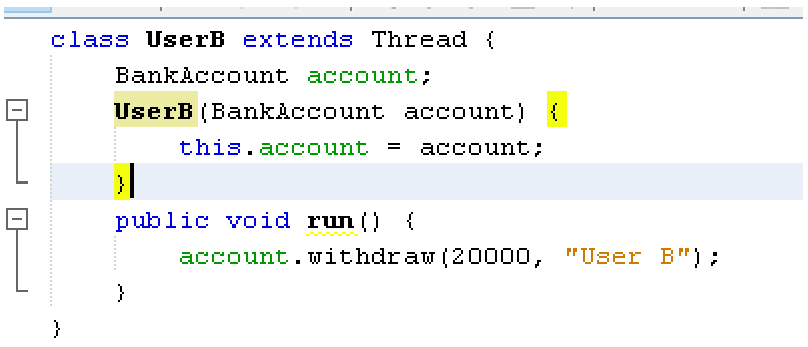
```java
class BankAccount {
    private int balance = 50000;
    synchronized void withdraw(int amount, String user) {
        System.out.println(user + " is attempting to withdraw Rs. " + amount);

        if (balance < amount) {
            System.out.println("Insufficient balance for " + user + ". Waiting for deposit...");
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println(e);
            }
        }
        if (balance >= amount) {
            System.out.println(user + " is proceeding with withdrawal of Rs. " + amount);
            balance -= amount;
            System.out.println(user + " completed withdrawal. Remaining balance: Rs. " + balance);
        } else {
            System.out.println(user + " could not withdraw due to insufficient funds.");
        }
    }
    synchronized void deposit(int amount, String user) {
        System.out.println(user + " is depositing Rs. " + amount);
        balance += amount;
        System.out.println("Updated balance after deposit by " + user + ": Rs. " + balance);
        notify();
    }
}
```

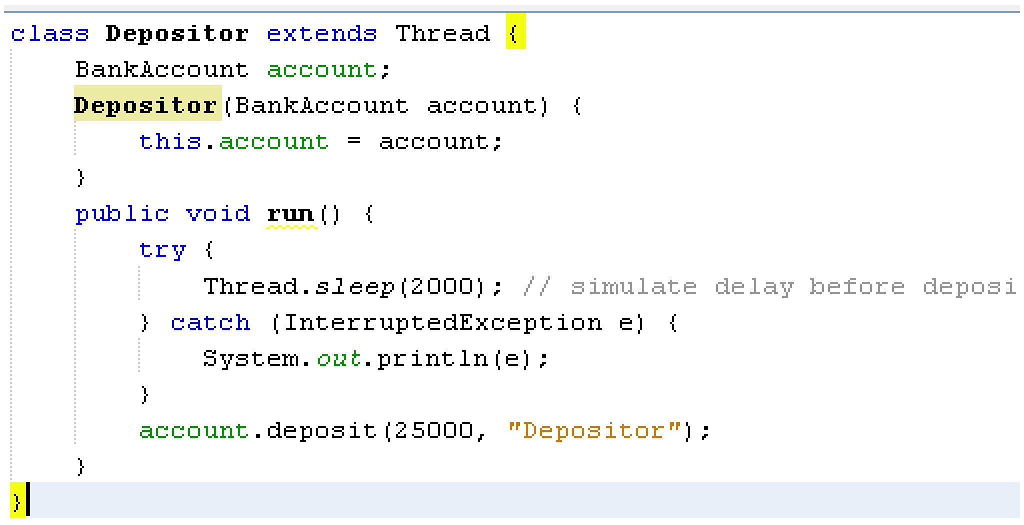
Aima khan
2023F-BSE-64



```java
class UserA extends Thread {
    BankAccount account;
    UserA(BankAccount account) {
        this.account = account;
    }
    public void run() {
        account.withdraw(45000, "User A");
    }
}
```

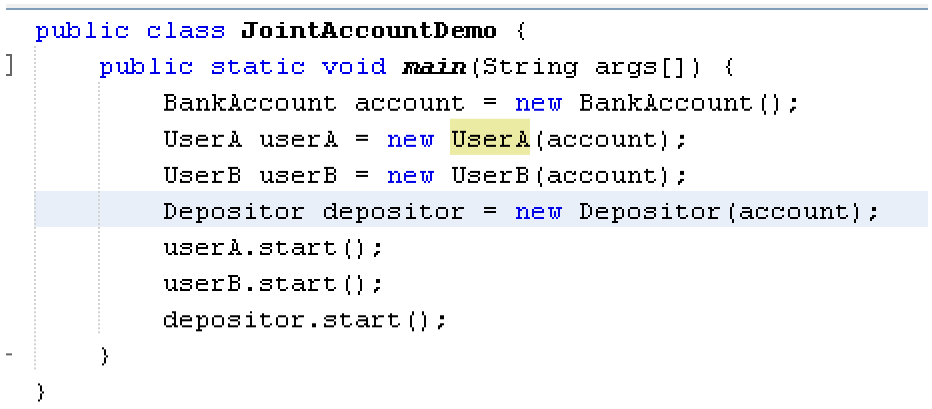


```java
class UserB extends Thread {
    BankAccount account;
    UserB(BankAccount account) {
        this.account = account;
    }
    public void run() {
        account.withdraw(20000, "User B");
    }
}
```



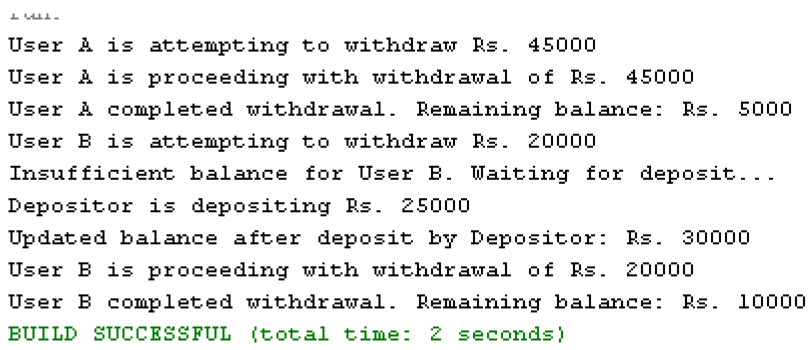
```java
class Depositor extends Thread {
    BankAccount account;
    Depositor(BankAccount account) {
        this.account = account;
    }
    public void run() {
        try {
            Thread.sleep(2000); // simulate delay before deposi
        } catch (InterruptedException e) {
            System.out.println(e);
        }
        account.deposit(25000, "Depositor");
    }
}
```

Aima khan
2023F-BSE-64



```
public class JointAccountDemo {
    public static void main(String args[]) {
        BankAccount account = new BankAccount();
        UserA userA = new UserA(account);
        UserB userB = new UserB(account);
        Depositor depositor = new Depositor(account);
        userA.start();
        userB.start();
        depositor.start();
    }
}
```

OUTPUT

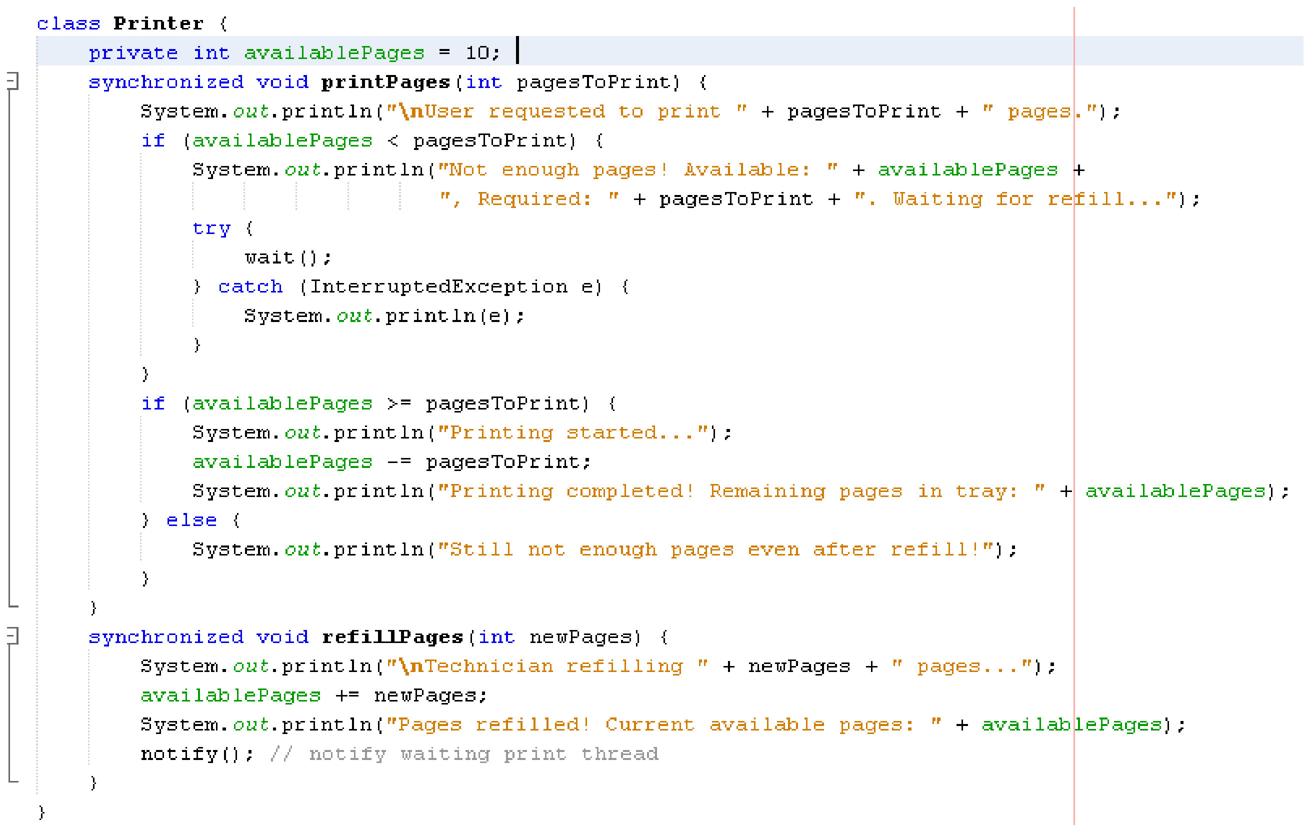


```
run:
User A is attempting to withdraw Rs. 45000
User A is proceeding with withdrawal of Rs. 45000
User A completed withdrawal. Remaining balance: Rs. 5000
User B is attempting to withdraw Rs. 20000
Insufficient balance for User B. Waiting for deposit...
Depositor is depositing Rs. 25000
Updated balance after deposit by Depositor: Rs. 30000
User B is proceeding with withdrawal of Rs. 20000
User B completed withdrawal. Remaining balance: Rs. 10000
BUILD SUCCESSFUL (total time: 2 seconds)
```

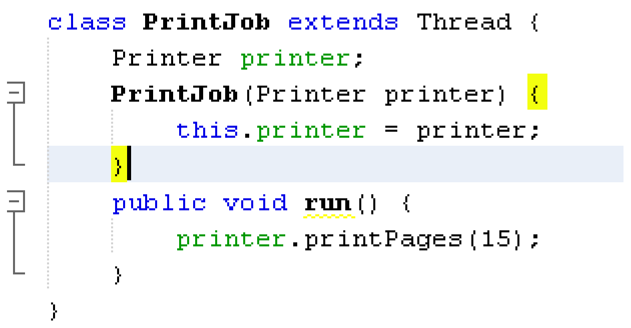
2. Create an inter thread communication program of printer job by implementing two threads, one for calculating the remaining pages in printer tray and other one will print the pages that are pending on queue. (Hint: If total pages are 10 and user sends job for 15 pages than print thread will be on wait and will be notified once available pages are equal or greater than printing pages).

CODE:

Aima khan
2023F-BSE-64

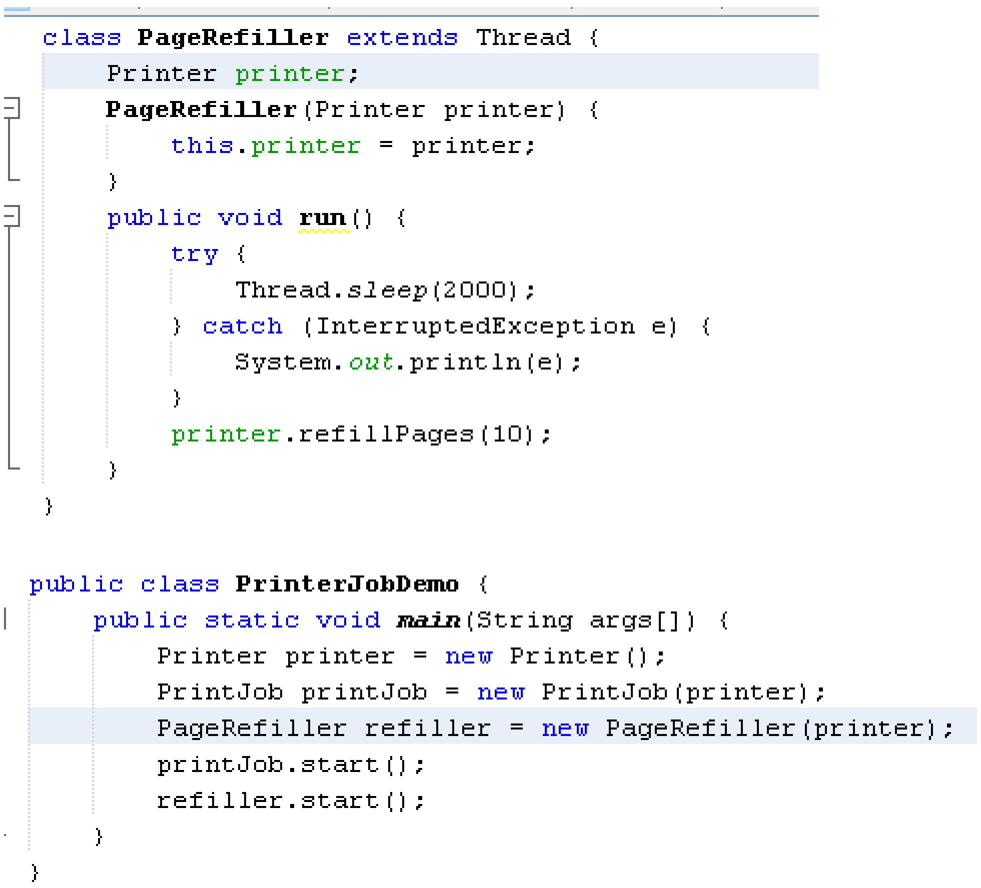


```java
class Printer {
    private int availablePages = 10;
    synchronized void printPages(int pagesToPrint) {
        System.out.println("\nUser requested to print " + pagesToPrint + " pages.");
        if (availablePages < pagesToPrint) {
            System.out.println("Not enough pages! Available: " + availablePages +
                    ", Required: " + pagesToPrint + ". Waiting for refill...");
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println(e);
            }
        }
        if (availablePages >= pagesToPrint) {
            System.out.println("Printing started...");
            availablePages -= pagesToPrint;
            System.out.println("Printing completed! Remaining pages in tray: " + availablePages);
        } else {
            System.out.println("Still not enough pages even after refill!");
        }
    }
    synchronized void refillPages(int newPages) {
        System.out.println("\nTechnician refilling " + newPages + " pages...");
        availablePages += newPages;
        System.out.println("Pages refilled! Current available pages: " + availablePages);
        notify(); // notify waiting print thread
    }
}
```



```java
class PrintJob extends Thread {
    Printer printer;
    PrintJob(Printer printer) {
        this.printer = printer;
    }
    public void run() {
        printer.printPages(15);
    }
}
```

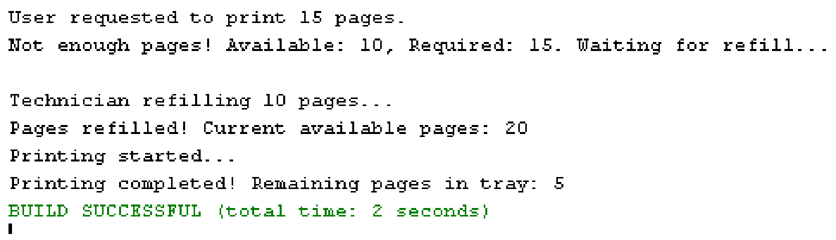
Aima khan
2023F-BSE-64



```java
class PageRefiller extends Thread {
    Printer printer;
    PageRefiller(Printer printer) {
        this.printer = printer;
    }
    public void run() {
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e) {
            System.out.println(e);
        }
        printer.refillPages(10);
    }
}

public class PrinterJobDemo {
    public static void main(String args[]) {
        Printer printer = new Printer();
        PrintJob printJob = new PrintJob(printer);
        PageRefiller refiller = new PageRefiller(printer);
        printJob.start();
        refiller.start();
    }
}
```

OUTPUT:



```
User requested to print 15 pages.
Not enough pages! Available: 10, Required: 15. Waiting for refill...

Technician refilling 10 pages...
Pages refilled! Current available pages: 20
Printing started...
Printing completed! Remaining pages in tray: 5
BUILD SUCCESSFUL (total time: 2 seconds)
```