LAB # 5

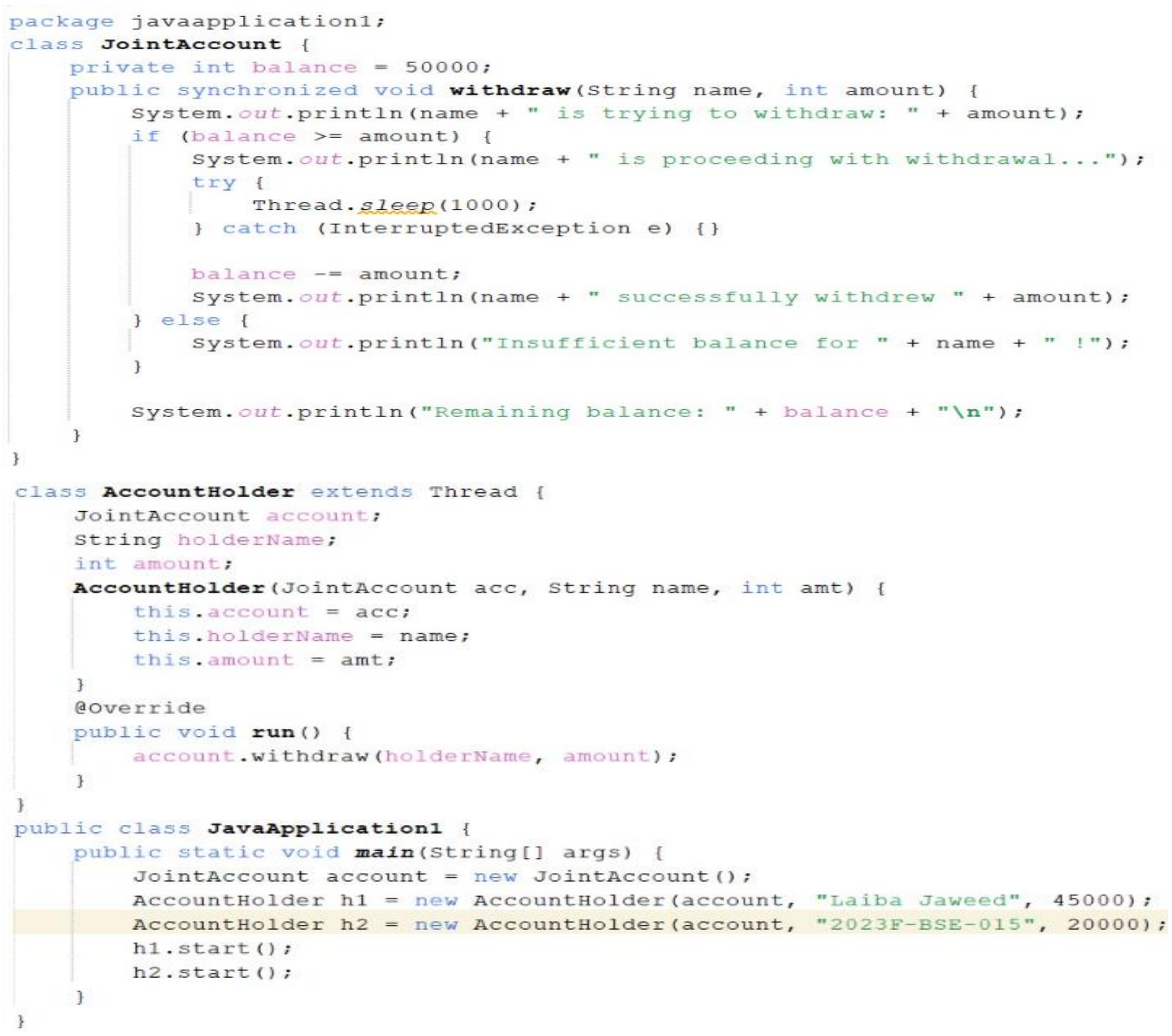
Inter-Thread Communication:

OBJECTIVE

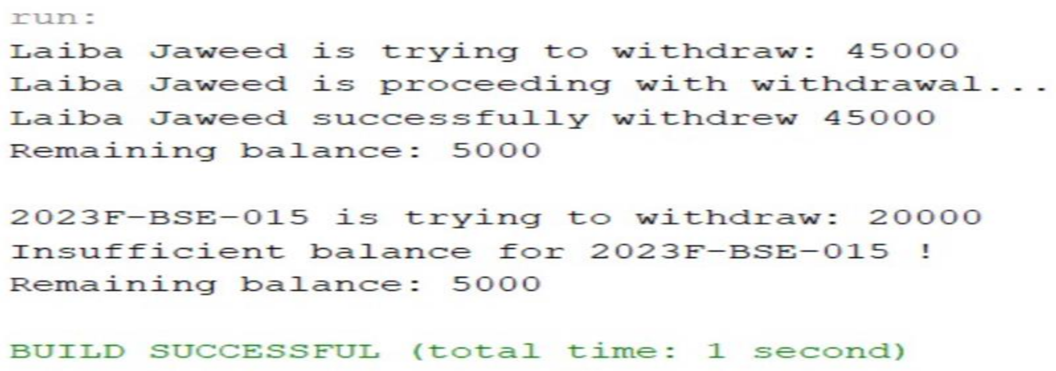
Develop an inter-thread user communication program by using synchronization.

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:

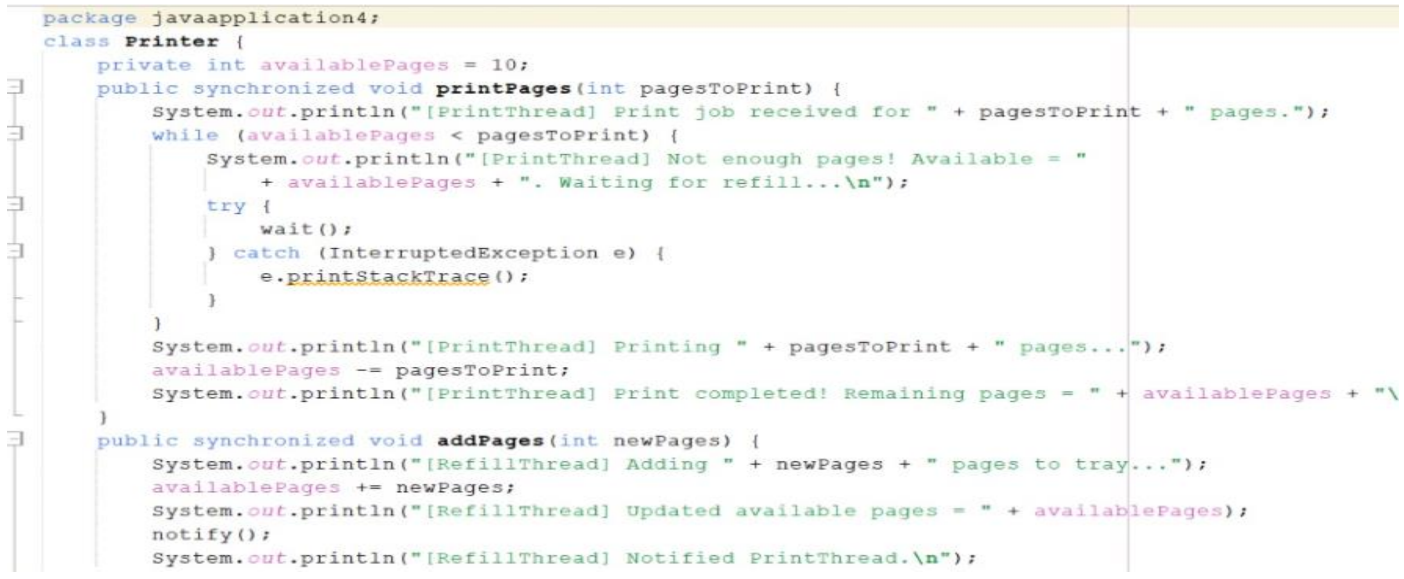


```java
package javaapplication1;
class JointAccount {
    private int balance = 50000;
    public synchronized void withdraw(String name, int amount) {
        System.out.println(name + " is trying to withdraw: " + amount);
        if (balance >= amount) {
            System.out.println(name + " is proceeding with withdrawal...");
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {}

            balance -= amount;
            System.out.println(name + " successfully withdrew " + amount);
        } else {
            System.out.println("Insufficient balance for " + name + " !");
        }

        System.out.println("Remaining balance: " + balance + "\n");
    }
}

class AccountHolder extends Thread {
    JointAccount account;
    String holderName;
    int amount;
    AccountHolder(JointAccount acc, String name, int amt) {
        this.account = acc;
        this.holderName = name;
        this.amount = amt;
    }
    @Override
    public void run() {
        account.withdraw(holderName, amount);
    }
}
public class JavaApplication1 {
    public static void main(String[] args) {
        JointAccount account = new JointAccount();
        AccountHolder h1 = new AccountHolder(account, "Laiba Jaweed", 45000);
        AccountHolder h2 = new AccountHolder(account, "2023F-BSE-015", 20000);
        h1.start();
        h2.start();
    }
}
```

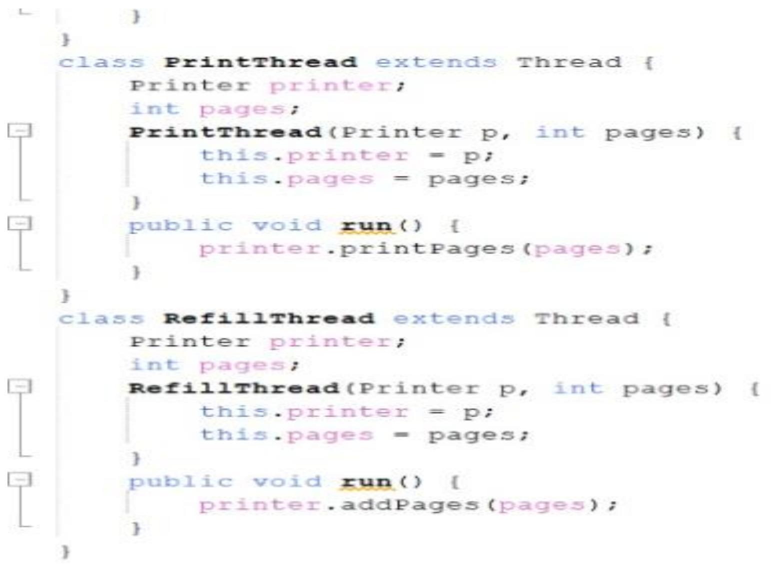


```
run:
Laiba Jaweed is trying to withdraw: 45000
Laiba Jaweed is proceeding with withdrawal...
Laiba Jaweed successfully withdrew 45000
Remaining balance: 5000

2023F-BSE-015 is trying to withdraw: 20000
Insufficient balance for 2023F-BSE-015 !
Remaining balance: 5000

BUILD SUCCESSFUL (total time: 1 second)
```

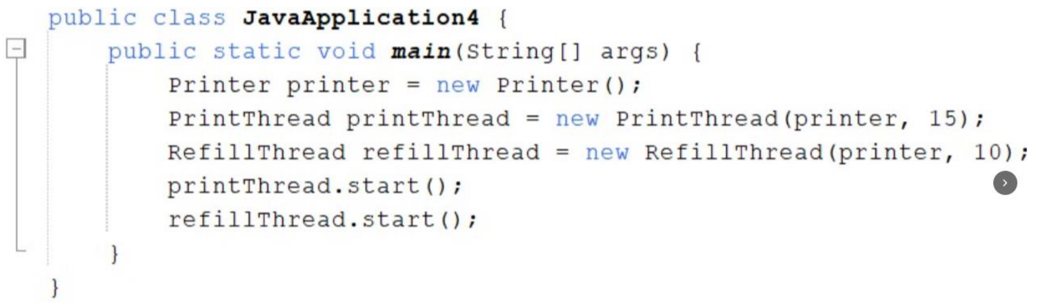
2. Create an inter thread communication program of printer job by implementing two threads, one for calculating the remaining pages in printer tray and other one will print the pages that are pending on queue. (Hint: If total pages are 10 and user sends job for 15 pages than print thread will be on wait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



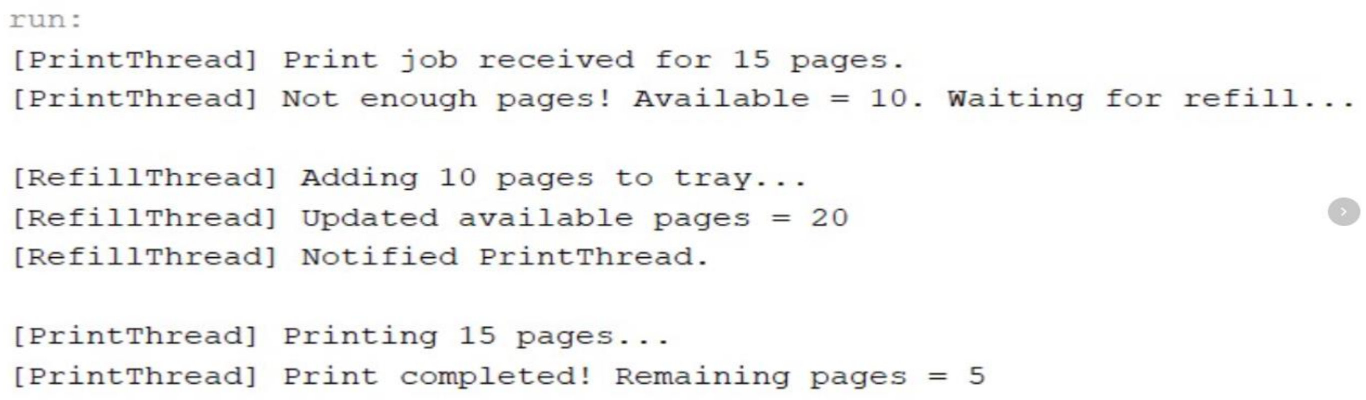
```java
package javaapplication4;
class Printer {
    private int availablePages = 10;
    public synchronized void printPages(int pagesToPrint) {
        System.out.println("[PrintThread] Print job received for " + pagesToPrint + " pages.");
        while (availablePages < pagesToPrint) {
            System.out.println("[PrintThread] Not enough pages! Available = "
                + availablePages + ". Waiting for refill...\n");
            try {
                wait();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        System.out.println("[PrintThread] Printing " + pagesToPrint + " pages...");
        availablePages -= pagesToPrint;
        System.out.println("[PrintThread] Print completed! Remaining pages = " + availablePages + "\
    }
    public synchronized void addPages(int newPages) {
        System.out.println("[RefillThread] Adding " + newPages + " pages to tray...");
        availablePages += newPages;
        System.out.println("[RefillThread] Updated available pages = " + availablePages);
        notify();
        System.out.println("[RefillThread] Notified PrintThread.\n");
```



```java
    }
}
class PrintThread extends Thread {
    Printer printer;
    int pages;
    PrintThread(Printer p, int pages) {
        this.printer = p;
        this.pages = pages;
    }
    public void run() {
        printer.printPages(pages);
    }
}
class RefillThread extends Thread {
    Printer printer;
    int pages;
    RefillThread(Printer p, int pages) {
        this.printer = p;
        this.pages = pages;
    }
    public void run() {
        printer.addPages(pages);
    }
}
```



```
public class JavaApplication4 {
    public static void main(String[] args) {
        Printer printer = new Printer();
        PrintThread printThread = new PrintThread(printer, 15);
        RefillThread refillThread = new RefillThread(printer, 10);
        printThread.start();
        refillThread.start();
    }
}
```



```
run:
[PrintThread] Print job received for 15 pages.
[PrintThread] Not enough pages! Available = 10. Waiting for refill...

[RefillThread] Adding 10 pages to tray...
[RefillThread] Updated available pages = 20
[RefillThread] Notified PrintThread.

[PrintThread] Printing 15 pages...
[PrintThread] Print completed! Remaining pages = 5
```