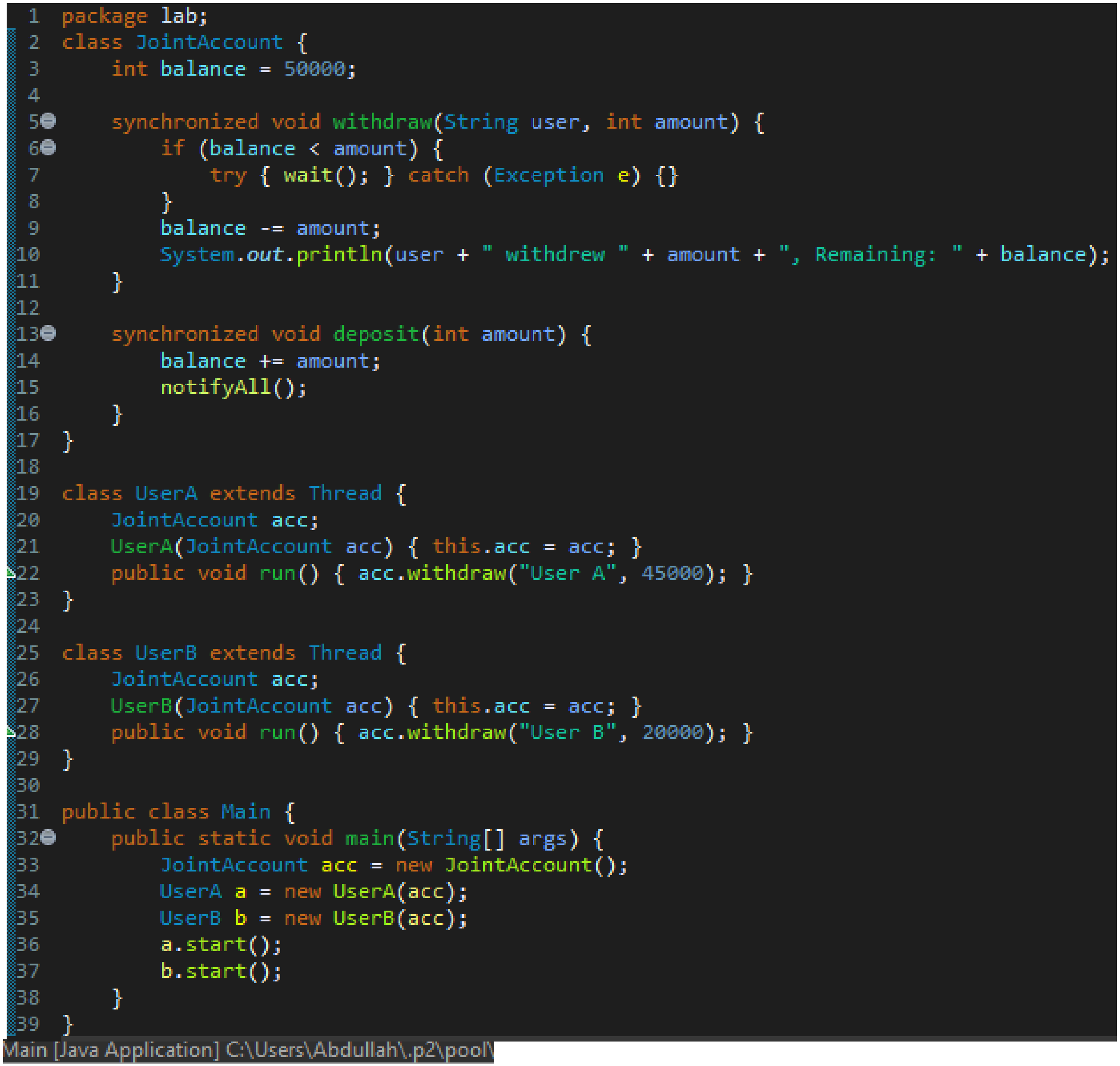
LAB 5

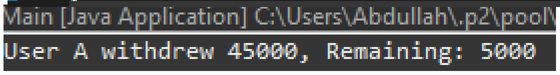
Inter-Thread Communication

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

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 A-Z.

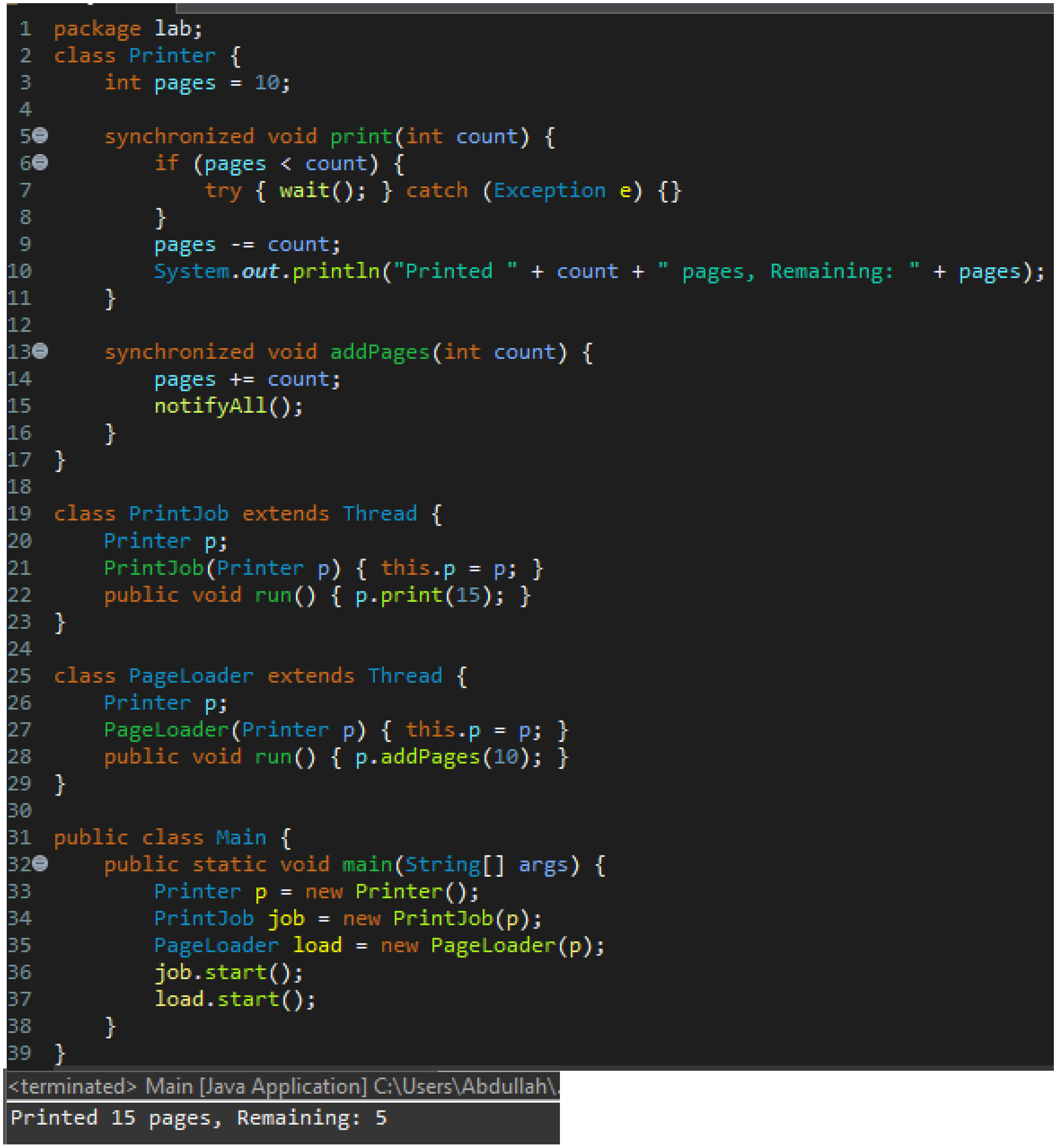


```java
package lab;
class JointAccount {
    int balance = 50000;

    synchronized void withdraw(String user, int amount) {
        if (balance < amount) {
            try { wait(); } catch (Exception e) {}
        }
        balance -= amount;
        System.out.println(user + " withdrew " + amount + ", Remaining: " + balance);
    }

    synchronized void deposit(int amount) {
        balance += amount;
        notifyAll();
    }
}

class UserA extends Thread {
    JointAccount acc;
    UserA(JointAccount acc) { this.acc = acc; }
    public void run() { acc.withdraw("User A", 45000); }
}

class UserB extends Thread {
    JointAccount acc;
    UserB(JointAccount acc) { this.acc = acc; }
    public void run() { acc.withdraw("User B", 20000); }
}

public class Main {
    public static void main(String[] args) {
        JointAccount acc = new JointAccount();
        UserA a = new UserA(acc);
        UserB b = new UserB(acc);
        a.start();
        b.start();
    }
}
```



```
User A withdrew 45000, Remaining: 5000
```

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 lab;
class Printer {
    int pages = 10;

    synchronized void print(int count) {
        if (pages < count) {
            try { wait(); } catch (Exception e) {}
        }
        pages -= count;
        System.out.println("Printed " + count + " pages, Remaining: " + pages);
    }

    synchronized void addPages(int count) {
        pages += count;
        notifyAll();
    }
}

class PrintJob extends Thread {
    Printer p;
    PrintJob(Printer p) { this.p = p; }
    public void run() { p.print(15); }
}

class PageLoader extends Thread {
    Printer p;
    PageLoader(Printer p) { this.p = p; }
    public void run() { p.addPages(10); }
}

public class Main {
    public static void main(String[] args) {
        Printer p = new Printer();
        PrintJob job = new PrintJob(p);
        PageLoader load = new PageLoader(p);
        job.start();
        load.start();
    }
}
```

```
<terminated> Main [Java Application] C:\Users\Abdullah\
Printed 15 pages, Remaining: 5
```