

Sau khi thực hành xong bài này, học viên có khả năng :

- Thực hành và sử dụng được Thread trong ứng dụng Java
- Ứng dụng được mô hình synchronize để đồng bộ các Thread .

Xây dựng ứng dụng Java Application thực hiện chức năng sau : thực hiện chuyển tiền từ tài khoản **a** sang tài khoản **b** thông qua các thread và theo mô hình đồng bộ (synchronize)

```
How much do you want tranfer (first) ?
10
How much do you want tranfer (second) ?
20
Tranfer processing : 1
The balance before tranfer:
Tom : $100
Jerry : $200
The balance amount after tranfer: 10
Tom : $90
Jerry : $210
Successful.
Tranfer processing : 2
The balance before tranfer:
Tom : $90
Jerry : $210
The balance amount after tranfer: 20
Tom : $70
Jerry : $230
Successful.
Tranfer completed.
```

Bước 1 . Mở ứng dụng NetBean , New Project | Java | Java Application , nhấn Next , trên hộp thoại “New Java Application ” điền các thông tin trong hộp thoại như sau :
Project Name : **ThreadInJava** ; project Location : “nhập đường dẫn lưu ứng dụng” và *Bỏ chọn* “Create Main Class” , nhấn Finish để thúc .

1.1 Nhấp phải chuột trên **<default package>** chọn New | Java Class , đặt tên lớp : **CustomerAccount**, nhấn Finish để kết thúc .

- 1.2 Nhấp phải chuột trên **<default package>** chọn New | Java Class , đặt tên lớp : **TransferManager**, nhấp Finish để kết thúc.
- 1.3 Nhấp phải chuột trên **<default package>** chọn New | Java Class, đặt tên lớp : **AccountManager**, nhấp Finish để kết thúc.

Bước 2. Phát triển các lớp để thực hiện các chức năng .

2.1 Phát triển lớp **CustomerAccount**, lớp này mô tả thông tin chi tiết cho tài khoản. Viết mã cho tập tin **CustomerAccount.java** như sau :

```
//Khai bao lop tai khoan khach hang
public class CustomerAccount {
    //ten tai khoan
    String accountName;
    //so du tai khoan
    int balance;
    public CustomerAccount(String accType, int bal) {
        accountName = accType;
        balance = bal;
    }
    //Khai bao phuong thuc lay ten tai khoan
    public String getAccountType() {
        return accountName;
    }
    //Khai bao phuong thuc lay so du tai khoan
    public int getBalance() {
        return balance;
    }
    //Khai bao phuong thuc cap nhat so du tai khoan
    public void setBalance(int amount) {
        balance = amount;
    }
}
```

2.2 Phát triển lớp **TransferManager**, lớp này thực hiện thao tác chuyển tiền và cập nhập số dư tài khoản thông qua một lớp trung gian là lớp **ExecuteTransfer**. Viết mã cho tập tin **TransferManager.java** như sau :

```
//Khai bao lop thuc hien chuyen tien theo co che dong bo
class ExecuteTranfer {
    //Khai bao phuong thuc chuyen tien theo co che dong bo giua 2 lan chuyen
    synchronized void Tranfer(Thread t ,CustomerAccount acc1,
        CustomerAccount acc2,int amount,int numTran )
    {
        try {
            System.out.println("Tranfer processing : "+numTran);
            //Tam ngưng 3s de gia lap |viec chuyen tien dang duoc xu ly
            Thread.sleep(3000);
            //Xuat so du tai khoan truoc khi chuyen
            System.out.println("The balance before tranfer: ");
            System.out.println(acc1.accountName+" : $" + acc1.getBalance());
            System.out.println(acc2.accountName+" : $" + acc2.getBalance());
            //Cap nhat so du tai khoan chuyen va tai khoan nhan
            acc1.setBalance(acc1.getBalance() - amount);
            acc2.setBalance(acc2.getBalance() + amount);
            //Xuat thong tin so du sau khi chuyen
            System.out.println("The balance amount after tranfer: "+amount);
            System.out.println(acc1.accountName+" : $" +acc1.getBalance());
            System.out.println(acc2.accountName+" : $" + acc2.getBalance());

        } catch (InterruptedException e) {
            System.out.println(e.getMessage());
        }

        System.out.println("Successful.");
    }
}

//Khai lop thread
public class TranferManager extends Thread {
    //acc1, acc2 luu tai khoan khach hang
    CustomerAccount acc1, acc2;
    //khai bao bien amount luu so tien can chuyen
    int amount;
    //Khai bao doi tuong objTran de thuc hien chuyen tien
    ExecuteTranfer objTran ;
    //numTran : luu so lan chuyen tien (lan 1 hoac lan 2)
    int numTran ;
}
```

```

public TranferManager(ExecuteTranfer tran, CustomerAccount a,
    CustomerAccount b, int m,int num) {
    acc1 = a;
    acc2 = b;
    amount = m;
    objTran = tran;
    numTran = num;
}
public void run(){
    objTran.Tranfer(this,acc1,acc2, amount,numTran);
}
}

```

2.3 Phát triển lớp **AccountManager**, lớp khởi tạo tài khoản , nhập số và kiểm tra số tiền cần chuyển,... . Viết mã cho tập tin **AccountManager.java** như sau :

```

import java.util.*;
public class AccountManager {
    //Khai bao phuong thuc nhap va kiem tra so tien can chuyen
    //phai nho hon so du trong tai khoan
    public static int InputAmount(CustomerAccount cus){
        int amount;
        Scanner input = new Scanner(System.in);
        do {
            amount = input.nextInt();
            if (amount > cus.getBalance())
                System.out.println("The tranfered amount should be less than "
                    + cus.getBalance());
        } while (amount > cus.getBalance());
        return amount;
    }

    public static void main(String[] args) throws InterruptedException {
        int amount;
        //Khai bao doi tuong objTran de lam tham so cho lop TranferManager(Thread)
        //thuc hien chuyen tien
        ExecuteTranfer objTran = new ExecuteTranfer();
        //Khoi tao tai khoan khach hang Tom voi so du : $100
        CustomerAccount tom = new CustomerAccount("Tom",100);
        //Khoi tao tai khoan khach hang Jerry voi so du : $200
        CustomerAccount jerry = new CustomerAccount("Jerry",200);
        //Nhap so tien chuyen lan 1
        System.out.println("How much do you want tranfer (first) ?");
        amount = InputAmount(tom);
    }
}

```

```

//Khoi tao thread thuc hien chuyen tien lan 1
TranferManager tranfer1 = new TranferManager(objTran,tom,jerry,amount,1);
//Nhap so tien chuyen lan 2
System.out.println("How much do you want transfer (second) ?");
amount = InputAmount(tom);
//Khoi tao thread thuc hien chuyen tien lan 1
TranferManager tranfer2 = new TranferManager(objTran,tom,jerry,amount,2);
//Thuc hien chuyen tien
tranfer1.start();
tranfer2.start();
//cac thread goi phuong thuc join de thread trong ham Main
//cho tat cac thread con (tranfer1,tranfer2) thuc hien hoan tat
//sau do moi thuc hien tiep (xuat chuoai : Tranfer completed.)
tranfer1.join();
tranfer2.join();
System.out.println("Tranfer completed.");
}

```

Bước 3: Chạy ứng dụng và kiểm tra kết quả

Nhập số tiền cần chuyển lần 1 và 2 , sau đó xem kết quả thực hiện . Mỗi lần chuyển tạm ngừng trong 3s .

```

How much do you want transfer (first) ?
10
How much do you want transfer (second) ?
20
Transfer processing : 1
The balance before tranfer:
Tom : $100
Jerry : $200
The balance amount after tranfer: 10
Tom : $90
Jerry : $210
Successful.
Transfer processing : 2
The balance before tranfer:
Tom : $90
Jerry : $210
The balance amount after tranfer: 20
Tom : $70
Jerry : $230
Successful.
Transfer completed.

```