

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

- Sử dụng được các lớp `FileInputStream`, `ObjectInputStream`,... để tạo tập tin nhị phân
- Ứng dụng được `serialization` để lưu trữ đối tượng vào trong tập tin nhị phân

Xây dựng ứng dụng Java Application thực hiện các chức năng sau : Lưu trữ thông tin các tài khoản vào trong tập tin `AccountManagement.dat`, kiểm tra điều kiện (số dư, số tiền rút, số tiền gửi), tìm thông tin của tài khoản dựa vào Account Number (số tài khoản), thực hiện thao tác rút tiền và gửi tiền.

***** Welcome to Merchants Bank *****

```
1. Create a new Account
2. Withdraw a cash
3. Deposit cash
4. Display account details
5. Exit
Enter your choice:
1
Create new account ...
Enter first name: tom
Enter last name: jerry
Enter account balance: 1000
```

```
1. Create a new Account
2. Withdraw a cash
3. Deposit cash
4. Display account details
5. Exit
Enter your choice:
2
Enter the account number: 1
Enter the amount you wish to withdraw: 300
Withdraw success...
Account Number: 1
Account Balance: 700
```

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 : **FileHandlingInJava** ; 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 : **Account**, nhấn Finish để kế thúc .
- 1.2 Nhấp phải chuột trên **<default package>** chọn New | Java Class , đặt tên lớp : **Bank**, nhấn Finish để kế thúc.

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

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

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

```
import java.io.*;
//Khai bao lop luu thong tin tai khoan
public class Account implements Serializable {
    int accountNumber;
    String lastName;
    String firstName;
    int accountBalance;
}
```

2.2 Phát triển lớp **Bank**, lớp này thực hiện các thao tác : lưu trữ danh sách các tài khoản , thêm , xem thông tin tài khoản, rút và gửi tiền trong tài khoản . Viết mã cho tập tin **Bank.java** như sau :

```
import java.util.*;
import java.io.*;
public class Bank {

    static int lastAccountNumber;
    //Khai bao mang luu tru danh sach cac tai khoan
    Account account[];
    Scanner input = new Scanner(System.in);
    File f = new File("AccountManagement.dat");
    public Bank() {
        //Khai bao mang account chua 10 tai khoan
        account = new Account[10];
        lastAccountNumber = 0;
        //Neu tap tin AccountManagement.dat da co thi doc danh sach
        // cac tai khoan vao trong mang account
    }
}
```

```
        if (f.exists()){
            try {
                FileInputStream fis = new FileInputStream(f);
                ObjectInputStream ois = new ObjectInputStream(fis);
                while (fis.available() > 0){
                    account[lastAccountNumber++] = (Account)ois.readObject();
                }
                ois.close();
                fis.close();
            } catch (Exception e){
                System.out.println("Exception occurred - " + e.getMessage());
            }
        }
    }
    //Khai bao phuong thuc tao account
    public void createAccount(Account objAccount){
        objAccount.accountNumber = ++lastAccountNumber;
        account[lastAccountNumber - 1] = objAccount;
    }
    //Khai bao phuong thuc rut tien
    public void withdraw(){
        int accNum = 0, amount = 0;
        //Nhap va kiem tra account number
        do {
            System.out.print("Enter the account number: ");
            accNum = input.nextInt();
            if (accNum <= 0 || accNum > lastAccountNumber)
                System.out.println("The account number is not valid.");

        }while (accNum <= 0 || accNum > lastAccountNumber);
        //Nhap va kiem tra so tien rut
        do {
            System.out.print("Enter the amount you wish to withdraw: ");
            amount = input.nextInt();
            if (amount < 100)
                System.out.println("The amount must be greater than 100");

        }while (amount < 100);
    }
}
```

```

//Kiem tra so tien con lai trong tai khoan toi thieu la 500
if (account[accNum - 1].accountBalance - amount < 500)
    System.out.println("The standard account must have minimum 500 in balance");
else {
    //Cap nhat so du trong tai khoan sau khi rut tien
    account[accNum - 1].accountBalance -= amount;
    System.out.println("Withdraw success...");
    System.out.println("Account Number: " + accNum);
    System.out.println("Account Balance: " + account[accNum - 1].accountBalance);
}
}
//Khao bao phuong thuc gui tien
public void deposit(){
    int accNum = 0, amount = 0;
    //Kiem tra so account number
    do {
        System.out.print("Enter the account number: ");
        accNum = input.nextInt();
        if (accNum <= 0 || accNum > lastAccountNumber)
            System.out.println("The account number is not valid.");
    }while (accNum <= 0 || accNum > lastAccountNumber);
    //Nhap va kiem tra so tien gui > 100
    do {
        System.out.print("Enter the amount you wish to deposit: ");
        amount = input.nextInt();
        if (amount < 100)
            System.out.println("The amount must be greater than 100");

    }while (amount < 100);
    //Cap nhat so du trong tai khoan
    account[accNum - 1].accountBalance += amount;
    System.out.println("Deposit success...");
    System.out.println("Account Number: " + accNum);
    System.out.println("Account Balance: " + account[accNum - 1].accountNumber);
}

```

```
//Khai bao phuong thuc hien thi thong tin cac tai khoan
public void display(){
    int accNum = 0;
    do {
        System.out.print("Enter the account number: ");
        accNum = input.nextInt();
        if (accNum <= 0 || accNum > lastAccountNumber)
            System.out.println("The account number is not valid.");

    }while (accNum <= 0 || accNum > lastAccountNumber);

    System.out.println("--- Account detail: ");
    System.out.println("Account Number: " + accNum);
    System.out.println("Last Name: " + account[accNum - 1].lastName);
    System.out.println("First Name: " + account[accNum - 1].firstName);
    System.out.println("Account Balance: " + account[accNum - 1].accountBalance);
}
}
```

2.3 Phát triển lớp **BankTest**, lớp này hiển thị một menu cho người dùng chọn các chức năng để thao tác : tạo tài khoản , xem tài khoản, rút và gửi tiền . Viết mã cho tập tin **BankTest.java** như sau :

```
import java.util.*;
import java.io.*;

public class BankTest {

    public static void main(String[] args) {
        Bank bank = new Bank();
        Scanner input = new Scanner(System.in);
        int choice;
        System.out.println("***** Welcome to Merchants Bank *****\n");
        while (true){
            System.out.println("1. Create a new Account");
            System.out.println("2. Withdraw a cash");
            System.out.println("3. Deposit cash");
            System.out.println("4. Display account details");
            System.out.println("5. Exit");
            do {
                System.out.println("Enter your choice: ");
                choice = input.nextInt();
            } while (choice < 1 || choice > 5);
        }
    }
}
```

```

switch (choice)
{
    case 1: System.out.println("Create new account ...");
        Account temp = new Account();
        do {
            input = new Scanner(System.in);
            System.out.print("Enter first name: ");
            temp.firstName = input.nextLine();
            input = new Scanner(System.in);
            System.out.print("Enter last name: ");
            temp.lastName = input.nextLine();
            input = new Scanner(System.in);
            System.out.print("Enter account balance: ");
            temp.accountBalance = input.nextInt();
            if (temp.accountBalance < 100)
                System.out.println("The account balance >= 100");
        } while (temp.accountBalance < 100);

        bank.createAccount(temp);
        break;

    case 2: bank.withdraw();
        break;
    case 3: bank.deposit();
        break;
    case 4: bank.display();
        break;

    //Khi user thoat ung dung ,luu thong tin cac tai khoan
    // trong mang account xuong AccountManagement.dat
    case 5: File f = new File("AccountManagement.dat");
        try{
            FileOutputStream fos = new FileOutputStream(f);
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            for (int i = 0; i < bank.lastAccountNumber; i++)
                oos.writeObject(bank.account[i]);
        }
}

```

```
        oos.close();
        fos.close();
        System.out.println("Thank you for using this program");
    } catch (IOException e) {
        System.out.println("IOException occurred - " + e.getMessage());
    }
    System.exit(0);
}
}
```

Bước 3: Chạy ứng dụng và kiểm tra kết quả

- Nhập vào số 1 : thêm tài khoản.
- Nhập vào số 2 : rút tiền trong tài khoản .
- Nhập vào số 3 : gửi tiền vào tài khoản
- Nhập vào số 4 : xem thông tin tài khoản
- Nhập vào số 5 : thoát ứng dụng.

```
***** Welcome to Merchants Bank *****
```

1. Create a new Account
2. Withdraw a cash
3. Deposit cash
4. Display account details
5. Exit

Enter your choice:

1

Create new account ...

Enter first name: tom

Enter last name: jerry

Enter account balance: 1000

```
1. Create a new Account
2. Withdraw a cash
3. Deposit cash
4. Display account details
5. Exit
Enter your choice:
2
Enter the account number: 1
Enter the amount you wish to withdraw: 300
Withdraw success...
Account Number: 1
Account Balance: 700
```