Lab 02: NGÂN HÀNG SỐ phiên bản v2.0.0

Tổng quan bài tập

Xây dựng phần mềm ngân hàng số cho phép kiểm tra mã số căn cước công dân từ đó biết được thông tin người cần truy cập, bao gồm các chức năng sau:

```
+----------+------------------------+----------+
| NGAN HANG SO | FX123456@v2.0.0               |
+----------+------------------------+----------+
 1. Them khach hang
 2. Them tai khoan cho khach hang
 3. Hien thi danh sach khach hang
 4. Tim theo ten khach hang
 5. Tim theo CCCD
 0. Thoat
+----------+------------------------+----------+
Chuc nang:
```

Mục tiêu giúp sinh viên hiểu được các vấn đề sau:

1. Vận dụng được biến, kiểu dữ liệu cơ bản trong Java.
2. Vận dụng được với câu điều kiện trong Java.
3. Vận dụng được với vòng lặp trong Java.
4. Vận dụng được với Scanner trong Java.
5. Vận dụng được với Collection ArrayList danh sách đối tượng.
6. Vận dụng được với hàm trong Java.
7. Vận dụng đối với class trong Java, biết định nghĩa class, khai báo đối tượng.
8. Vận dụng được tính kế thừa, đóng gói trong lập trình hướng đối tượng.

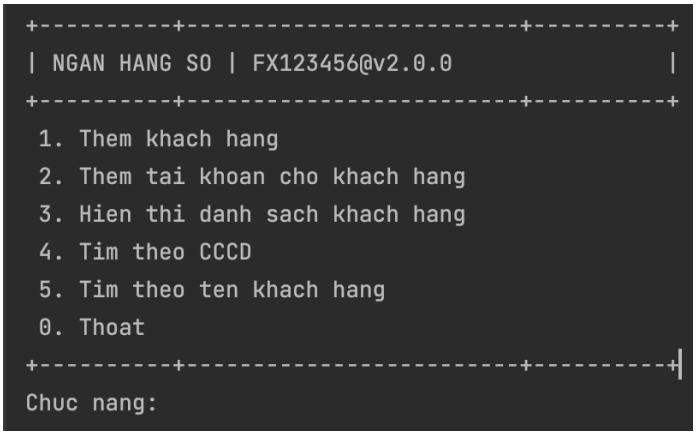
1. Mô tả

- Phần mềm ngân hàng số cho phép quản lý khách hàng, tài khoản khách hàng, tìm kiếm thông tin khách hàng thêm tên hoặc số căn cước công dân.

- Ứng dụng cần đảm bảo các chức năng và yêu cầu cơ bản tuy nhiên bạn có thể thêm chức năng bổ sung vào theo ý thích.

2. Thiết kế giao diện

- Yêu cầu giao diện dạng console đơn giản.
- Hiển thị thông tin thành khối rõ ràng và dễ nhìn.



3. Yêu cầu chức năng cơ bản

- Khi chương trình được khởi chạy đầu tiên sẽ hiển thị mô tả ngắn gọn về chương trình để người dùng hiểu được hệ thống này sẽ hoạt động thế nào, có chức năng gì ví dụ như hiển thị tên phần mềm viết in hoa:

  NGAN HANG SO

- Hiển thị tên tác giả (mã số sinh viên) và phiên bản (version) phần mềm theo mẫu

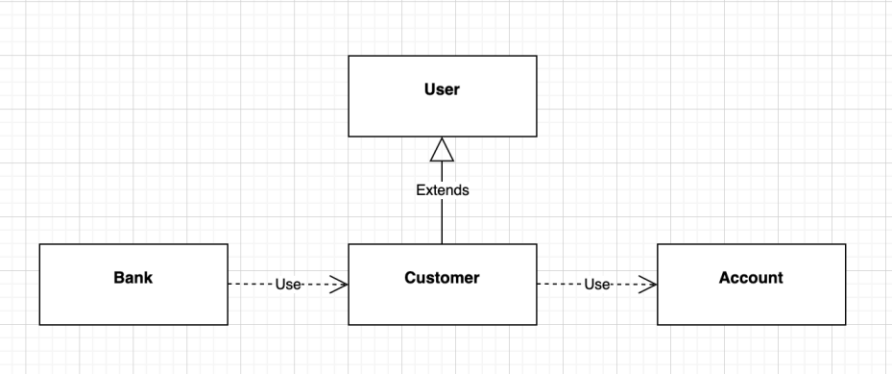
  2012111@v2.0.0

Trong đó:

  - **2012111**: là mã số sinh viên.
  - 2.0.0: Phiên bản phần mềm.
    - Hiển thị dường phân cách giữa tiêu đề và nội dung chức năng.
    - Hiển thị menu cho người dùng chọn theo mẫu ví dụ.
      - 1. Chức năng thêm khách hàng.
      - 2. Chức năng thêm tài khoản cho khách hàng.

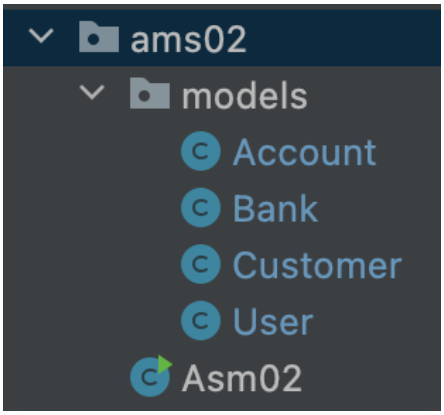
- 3. Hiển thị danh sách khách hàng.
- 4. Tìm theo tên khách hàng.
- 5. Tìm theo CCCD.
- 0. Thoát

4. Gợi ý mô hình quan hệ các lớp của bài tập này (tham khảo)



5. Tổ chức code

Tổ chức lưu trữ như hình dưới đây.



- Tạo **folder asm02** dùng để chứa source code của bài tập 02, theo hình trên.
- File **User.java**: định nghĩa cho class quản lý thông tin người dùng của hệ thống.
- File **Customer.java**: định nghĩa cho class quản lý thông tin khách hàng.
- File **Account.java**: định nghĩa cho class quản lý thông tin tài khoản khách hàng.
- File **Bank.java**: định nghĩa cho class quản lý thông tin ngân hàng.

6. Mô tả chi tiết

- Tạo **file User.java** chứa thông tin người dùng của hệ thống gồm các thuộc tính sau:
    - name (String): chứa thông tin tên của người dùng.
    - customerId: mã căn cước công dân định danh của khách hàng, là chuỗi 12 ký tự toàn số **(thoả mãn điều kiện hợp lệ của assignment 1).**
    - setCustomerId(): phương thức setter cho CCCD của khách hàng, chỉ cập nhật khi dữ liệu CCCD mới hợp lệ **(thoả mãn điều kiện hợp lệ của assignment 1)**
    - getCustomerId (): phương thức getter cho CCCD của khách hàng.
    - getName(), setName(): phương thức getter/setter cho thuộc tính name.

```java
public abstract class User {

    private String name;
    private String customerId;

    public User() {
    }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public String getCustomerId() { return customerId; }

    public void setCustomerId(String customerId) {...}
}
```

- Tạo file Account.java chứa thông tin quản lý tài khoản của khách hàng gồm:
- accountNumber (String): số tài khoản khách hàng gồm 6 chữ số, không được trùng nhau trong hệ thống ngân hàng (ví dụ: **100009**).
- balance (double): số dư của tài khoản.
- getBalance(), setBalance(): phương thức getter/setter cho thuộc tính balance.
- getAccountNumber(), setAccountNumber (): phương thức getter/setter cho thuộc tính **accountNumber**.

- isPremium(): phương thức cho biết tài khoản có phải là premium hay không, 1 tài khoản là premium nếu như số dư (balance) tối thiểu **10_000_000 VNĐ**
- Hàm **toString()**, xuất thông tin tài khoản gồm mã tài khoản và số dư, tham khảo ví dụ sau:

```
123456 |                                10,000,000đ
```

- Tạo **file Customer.java** chứa thông tin khách hàng của ngân hàng gồm:
    - Kế thừa lại class **User**.
    - **accounts: List<Account>**: thuộc tính accounts dạng List<Account>, lưu danh sách các tài khoản của khách hàng, có thể thêm, xoá tài khoản tuỳ ý.
    - **getAccounts():** phương thức getAccounts() để lấy danh sách các tài khoản của khách hàng.
    - **isPremium():** phương thức cho biết khách hàng có phải là premium hay không, 1 khách hàng là premium nếu có ít nhất 1 tài khoản (Account) là premium.
    - **addAccount():** phương thức có tham số là **newAccount: Account**, dùng để thêm tài khoản mới cho khách hàng. Tài khoản được thêm khi và chỉ khi số tài khoản chưa từng tồn tại trước đó.
    - Hàm **getBalance()** dùng để tính toán số dư của khách hàng là tổng số dư của tất cả tài khoản mà khách hàng có.
    - Hàm **displayInformation** dùng để hiển thị cho đối tượng Customer gồm thông tin chi tiết tài khoản với:
        - Hàng đầu tiên là thông tin tài khoản gồm: CCCD, tên, loại khách hàng nếu là Premium thì in ra Premium, ngược lại là Normal, sau cùng là tổng số tiền mà khách hàng này có.
        - Các hàng tiếp theo là các tài khoản mà khách hàng này sở hữu, được đánh số thứ tự, mã tài khoản và số dư của riêng từng tài khoản.

```
001215000001  |        FUNIX | Premium |  11,100,000đ
1      123456 |                           1,000,000đ
2      234567 |                          10,000,000đ
3      345678 |                             100,000đ
```

- Tạo **file Bank.java** chứa thông tin ngân hàng số
    - **Id: (String)**: chứa mã định danh của ngân hàng, được khởi tạo giá trị tự động khi đối tượng được khởi tạo trong constructor. **(xem thêm phần tạo id ngẫu nhiên)**
    - **customers: List<Customer>**: thuộc tính customers dạng List<Customer>, lưu danh sách tất cả khách hàng của ngân hàng, có thể thêm, xoá khách hàng tuỳ ý.
    - **addCustomer()**: phương thức có tham số là **newCustomer: Customer**, dùng để thêm khách hàng mới cho ngân hàng. Khách hàng được thêm khi và chỉ khi số CCCD của khách hàng chưa từng tồn tại trước đó.
    - **isCustomerExisted ()**: phương thức có tham số là **customerId**, là số CCCD, dùng để kiểm tra xem khách hàng này đã tồn tại trong ngân hàng hay chưa. Nếu tồn tại trả về true, không tồn tại trả về false.
    - **getCustomers()**: phương thức **getCustomers()** để lấy danh sách tất cả khách hàng của ngân hàng.

- Tạo file **Asm02.java** chứa hàm main chương trình chính.
    - Khởi tạo biến Bank dùng chung cho toàn bộ bài tập

```java
public class Asm02 {

    private static final Bank bank = new Bank();

    public static void main(String[] args) {

    }
```

- Phân chia hàm hợp lý, không được để hết code vào hàm main.
- Toàn bộ thao tác các chức năng thông qua đối tượng **bank**.

7. Cách tạo id ngẫu nhiên

- Trong hàm khởi tạo (constructor), dùng hàm UUID.randomUUID() của Java để tạo chuỗi ngẫu nhiên và gán cho id. Tham khảo ví dụ mẫu

```java
public Bank() {
    this.customers = new ArrayList<>();
    this.id = String.valueOf(UUID.randomUUID());
}
```

8. Yêu cầu chức năng cơ bản

- **Chức năng 1: Thêm khách hàng vào ngân hàng**
    - Thêm được tên khách hàng.
    - Thêm được số CCCD **(thỏa mãn điều kiện hợp lệ của assignment 1)**, nếu không thỏa thì yêu cầu nhập lại.

```
+----------+-------------------------+----------+
| NGAN HANG SO | FX123456@v2.0.0                |
+----------+-------------------------+----------+
 1. Them khach hang
 2. Them tai khoan cho khach hang
 3. Hien thi danh sach khach hang
 4. Tim theo CCCD
 5. Tim theo ten khach hang
 0. Thoat
+----------+-------------------------+----------+
Chuc nang: 1
Nhap ten khach hang:
FUNIX
Nhap so CCCD:
123456
Nhap so CCCD:
001215000001
```

- **Chức năng 2: Thêm tài khoản cho khách hàng**
    - Yêu cầu xác nhận CCCD khách hàng nào, trước khi thêm tài khoản cho khách hàng, nếu CCCD không tồn tại trong hệ thống ngân hàng thì yêu cầu nhập lại.
    - Nhập mã số khách hàng gồm 6 chữ số.
    - Nhập số dư tài khoản khách hàng, lưu ý số dư của tài khoản, không được nhỏ hơn **50_000 VNĐ**. Ngược lại báo lỗi cho người dùng nhập lại.

```
Chuc nang: 2
Nhap CCCD khach hang:
001215000001
Nhap ma STK gom 6 chu so:
123456
Nhap so du:
50000000
```

- **Chức năng 3: Hiển thị danh sách khách hàng**
    - Dòng 1: hiển thị thông tin theo định dạng gồm có số CCCD, tên, số dư và loại khách hàng **(1 khách hàng là premium nếu có ít nhất 1 tài khoản (Account) là premium)**
    - Dòng 2: Những dòng tiếp theo đánh số thứ tự các tài khoản của khách hàng gồm cố mã số tài khoản và số dư

```
037456789111 |       David |   Normal |   1,000,000đ
1      123456 |                          1,000,000đ
037456789112 |        Evan |  Premium |  12,000,000đ
1      234567 |                         10,000,000đ
2      345678 |                          2,000,000đ
```

- **Chức năng 4: Tìm theo CCCD khách hàng.**
    - Nhập CCCD khách hàng, yêu cầu tìm chính xác.
    - Hiển thị theo định dạng của chức năng 3.

9. Yêu cầu chức năng nâng cao

- Cài đặt hàm **setCustomerId**: Kiểm tra điều kiện hàm đúng CCCD thì mới gán giá trị, ngược lại quăng lỗi Exception
- Cài đặt hàm **addAccount** trong **class Customer**: chỉ thêm những account mà khách hàng chưa có.
- Cài đặt hàm **addAccount** trong class Bank có 2 tham số customerId, account. Chỉ thêm những account vào khách hàng có trong bank, tận dụng hàm addAccount của Customer
- **Chức năng 5: Tìm theo tên khách hàng**
    - Nhập từ khoá tìm theo tên khách hàng, yêu cầu tìm gần đúng.
    - Hiển thị theo định dạng của chức năng 3.

10. Hướng dẫn nộp bài

- Nén toàn bộ thư mục và các tài nguyên cần thiết của dự án để giảng viên chấm điểm bằng file .zip
- Chọn File -> Export -> Project to Zip file.