

1. Phân Tích Hệ Thống (OOA)

Đối Tượng (Objects)

Các đối tượng chính trong hệ thống quản lý tài khoản ngân hàng bao gồm:

1. **Account:** Đại diện cho tài khoản ngân hàng chung.
2. **SavingsAccount:** Lớp con kế thừa từ Account, đại diện cho tài khoản tiết kiệm có lãi suất.
3. **Customer:** Đại diện cho khách hàng của ngân hàng, có thể sở hữu nhiều tài khoản.
4. **Transaction:** Đại diện cho một giao dịch tài chính, bao gồm các giao dịch nạp tiền, rút tiền và chuyển tiền.

Thuộc Tính (Attributes)

Mỗi đối tượng có các thuộc tính riêng để mô tả các đặc điểm của chúng:

- **Account:**
 - **ownerName:** Tên chủ tài khoản.
 - **accountNumber:** Số tài khoản.
 - **balance:** Số dư tài khoản.
 - **history:** Lịch sử giao dịch của tài khoản.
- **SavingsAccount (Kế thừa từ Account):**
 - **interestRate:** Lãi suất áp dụng cho tài khoản tiết kiệm.
- **Customer:**
 - **name:** Tên khách hàng.
 - **customerID:** ID của khách hàng.
 - **accounts:** Danh sách các tài khoản của khách hàng.
- **Transaction:**
 - **amount:** Số tiền trong giao dịch.
 - **type:** Loại giao dịch (nạp tiền, rút tiền, chuyển tiền).
 - **date:** Ngày thực hiện giao dịch.

Phương Thức (Methods)

Mỗi đối tượng có các phương thức để thao tác với các thuộc tính của chúng:

- **Account:**
 - **deposit(amount):** Nạp tiền vào tài khoản.
 - **withdraw(amount):** Rút tiền từ tài khoản.
 - **displayInfo():** Hiển thị thông tin tài khoản.
- **SavingsAccount:**
 - **applyInterest():** Áp dụng lãi suất cho tài khoản tiết kiệm.
 - **withdraw(amount):** Rút tiền từ tài khoản tiết kiệm (có các hạn chế như không rút quá giới hạn hoặc không cho phép thâm hụt).
- **Customer:**
 - **addAccount(account):** Thêm tài khoản vào danh sách tài khoản của khách hàng.
 - **displayAccounts():** Hiển thị thông tin tất cả các tài khoản của khách hàng.
- **Transaction:**
 - **display():** Hiển thị thông tin giao dịch.
 - **getTypeString():** Lấy kiểu giao dịch dưới dạng chuỗi.

Quan Hệ Kế Thừa (Inheritance)

- **SavingsAccount kế thừa từ Account.**
 - Lớp **SavingsAccount** thừa hưởng các thuộc tính và phương thức của **Account** như **balance**, **deposit()**, **withdraw()**, **displayInfo()**.
 - Lớp **SavingsAccount** thêm một thuộc tính mới là **interestRate** và phương thức **applyInterest()**.
 - Lớp **SavingsAccount** cũng ghi đè phương thức **withdraw()** để áp dụng các hạn chế đặc biệt cho tài khoản tiết kiệm.

Nạp chồng Toán Tử (Operator Overloading)

- **Toán tử +=:** Được nạp chồng để cho phép thêm giao dịch vào tài khoản. Nếu giao dịch là DEPOSIT, số tiền sẽ được nạp vào tài khoản, nếu là WITHDRAWAL, số tiền sẽ bị rút.
 - **Toán tử ==:** Được nạp chồng để so sánh hai tài khoản dựa trên số dư của chúng. Nếu số dư của hai tài khoản giống nhau, toán tử trả về true.
-

2. Giải Thích Mã Nguồn (Code Walkthrough)

Lớp Account

Lớp Account là lớp cơ sở (base class) dùng để quản lý các tài khoản ngân hàng. Lớp này có các phương thức để thực hiện các hành động cơ bản như nạp tiền (deposit), rút tiền (withdraw), và hiển thị thông tin tài khoản (displayInfo). Các giao dịch cũng được lưu trữ trong một lịch sử giao dịch (history).

Lớp SavingsAccount

Lớp SavingsAccount kế thừa từ lớp Account và thêm một thuộc tính interestRate để tính lãi suất. Lớp này cũng ghi đè phương thức withdraw() để không cho phép rút quá số tiền có trong tài khoản hoặc vượt quá giới hạn rút tiền cho phép.

Phương thức applyInterest() được sử dụng để tính và áp dụng lãi suất cho tài khoản tiết kiệm.

Lớp Customer

Lớp Customer dùng để lưu trữ thông tin của khách hàng và các tài khoản của họ. Mỗi khách hàng có thể sở hữu nhiều tài khoản ngân hàng. Các phương thức trong lớp này cho phép thêm tài khoản mới vào danh sách và hiển thị thông tin các tài khoản của khách hàng.

Nạp Chồng Toán Tử

- **Toán tử +=** cho phép thêm giao dịch vào tài khoản. Khi một giao dịch được thêm vào tài khoản, số tiền sẽ được nạp hoặc rút theo loại giao dịch.
 - **Toán tử ==** so sánh hai tài khoản dựa trên số dư của chúng.
-

3. Kết Quả Kiểm Thử (Test Results)

Dưới đây là một số ví dụ về các thao tác trong hệ thống và kết quả của chúng:

Ví Dụ 1: Tạo Tài Khoản và Nạp Tiền

- **Input:** Nạp \$100 vào tài khoản có số dư ban đầu là \$500.
- **Output:**
- **Account #1001 | Owner: Nguyen Van A | Balance: \$600**

Ví Dụ 2: Rút Tiền và Kiểm Tra Lịch Sử Giao Dịch

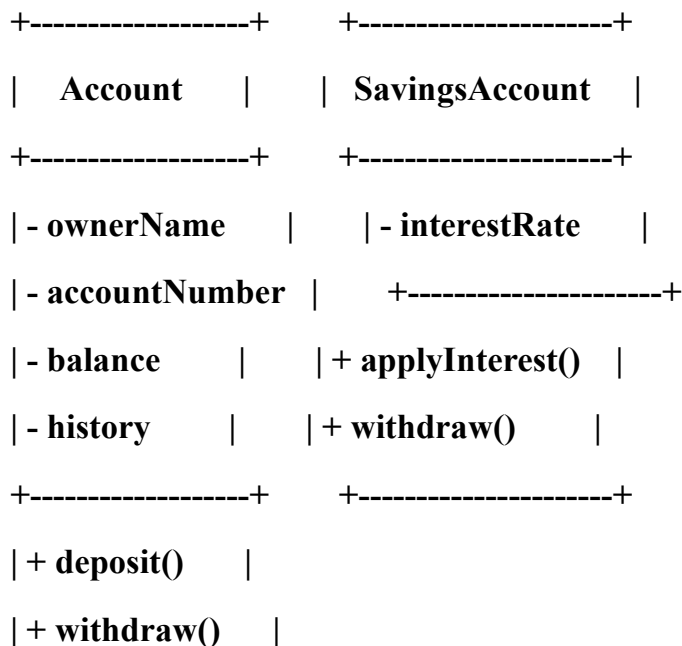
- **Input:** Rút \$200 từ tài khoản có số dư \$600.
- **Output:**
- **Account #1001 | Owner: Nguyen Van A | Balance: \$400**

Ví Dụ 3: Áp Dụng Lãi Suất Cho Tài Khoản Tiết Kiệm

- **Input:** Áp dụng lãi suất 3.5% cho tài khoản tiết kiệm có số dư \$10000.
- **Output:**
- **Interest of \$350 applied.**
- **Account #2001 | Owner: Le Van C | Balance: \$10350**

4. Sơ Đồ UML

Sơ Đồ Lớp UML



| + **displayInfo()** |

+-----+

^

|

Inheritance

|

+-----+

| **Transaction** |

+-----+

| - **amount** |

| - **type** |

| - **date** |

+-----+

| + **display()** |

| + **getTypeString()** |

+-----+

Sơ Đồ Tuần Tự UML (Sequence Diagram)

Customer Account Transaction

| | |

| **deposit()** | |

|----->| |

| | **create** |

| | **Transaction** |

| | <-----|

| | **display()** |

| |----->|

		updateBalance

5. Sử Dụng Mô Hình AI (LLM Usage)

Trong quá trình phát triển hệ thống này, tôi đã sử dụng ChatGPT để hỗ trợ trong việc:

- **Lên ý tưởng về việc nạp chồng toán tử:** Tôi đã hỏi ChatGPT về cách nạp chồng toán tử += để thêm giao dịch vào tài khoản và == để so sánh tài khoản.
- **Hỗ trợ trong việc kiểm tra lỗi:** ChatGPT giúp tôi nhận diện một số lỗi tiềm ẩn trong mã nguồn và đưa ra các giải pháp sửa lỗi.