

간단한 은행 계좌 관리 시스템 구현

박규태 | 010-2795-4422 | rbxo5872@naver.com





TABLE OF CONTENTS

01 프로젝트 개요 및 목표

Q

02 DB 설계 및 연동 구조

Q

03 시스템 전체 구조 및 클래스 설계

Q

04 계좌 기능 구현 과정 (입금, 출금, 이체)

Q

05 실제 구현 모습

Q

06 구현 중 겪은 문제점과 해결 방안

 \mathcal{Q}



CHAPTER 01 Q

프로젝트 개요 및 목표

1. 자바 문법 학습 후 실습용으로 은행 기초관리시스템을 기획

2

2. 클래스, 참조, 생성자 등 자바 문법의 실제 적용을 목표로 설정

 \mathcal{L}

3. 콘솔 기반 계좌 기능 구현을 통해 객체지향 구조를 연습함

2



0

CHAPTER 02 Q

Account 테이블 설명

KEY POINTS

계좌 정보 테이블

은행 시스템의 가장 핵심이 되는 고객 계좌 정보를 저장하는 테이블입니다. 각 계좌는 account_id를 기준으로 고유하게 식별되며,고객명 (customer_name), 잔액(balance), 생성일시(created_at) 정보를 포함합니다.이 테이블은 입출금 및 거래 이력과 연결되며, 모든 금융 활동의중심 역할을합니다.정규화된 구조로 설계되어, 고객 단위로 계좌를 효율적으로 관리할 수 있습니다.

#기본정보

#정규화

#잔액관리





CHAPTER 02 Q

Transaction 테이블 설명

KEY POINTS

거래 내역 테이블

각 계좌에서 발생한 입금, 출금, 이체 등의 모든 거래 정보를 기록하는 테이블입니다. transaction_id로 거래를 식별하고, 어떤 계좌에서 발생했는지를 account_id로 연결(외래키)합니다. 거래 유형(type), 금액 (amount), 발생 일시(transaction_date) 정보를 포함합니다. 이 테이블 덕분에 거래 이력을 추적하고, 이상 거래나 잔액 계산 등을 처리할 수 있게 됩니다.

#입출금 #이체 #거래추적



CHAPTER 03 Q 시스템 전체 구조 및 클래스 설계

STEP.01

사용자 입력

- 콘솔을 통해 메뉴 선택(조회, 생성 등)을 입력받음
- 사용자의 입력값을 바탕으로 BankService 메서드 호출.
- 프로그램의 진입 지점이자 모 든 흐름의 시작점

~

STEP.02

서비스 로직 처리

- 사용자 요청에 따라 입금, 출금, 이 체, 조회 등 비즈니스 로직 실행
 - 각 기능은 내부적으로 DAO를 호출 해 DB와 연동
- 예외 처리와 유효성 검사(계좌 존재 여부, 비밀번호 일치 등)를 수행

STEP.03

DAO를 통한 DB 접근

- JDBC를 활용하여 실제 SQL 실행
- 계좌 생성, 잔액 업데이트, 계좌 목록 조회 등 직접적인 DB 조 작 수행

STEP.04

결과

- 입금/출금/이체 같은 중요한 작 업을 안전하게 처리하기 위해 트랜잭션을 사용했습니다.
- 이 방식은 데이터가 잘못 저장 되거나 엉키는 걸 막기 위해 꼭 필요한 처리 방식입니다.

이 흐름은 MVC의 컨셉을 간단히 반영한 구조로, 각 역할을 분리해 유지보수가 쉽도록 설계되었습니다.



콘솔 기반 실제 구현 모습

```
=== 뱅크 시스템 ===
1. 계좌 조회
2. 입금
3. 출금
4. 이체
5. 전체 계좌 목록
6. 계좌 생성
0. 종료
메뉴 선택 >
```

```
메뉴 선택 > 1
조회할 계좌 ID: 1
비밀번호 4자리: 1111
계좌번호: 1
고객명: 홍길동
잔액: 127000.00
개설일: 2025-06-25T10:11:27
```

콘솔 화면을 통해 입금, 출금, 이체 등의 기능을 직접 실행할 수 있습니다. 위 예시는 계좌 조회 기능이 정상적으로 작동하는 모습을 보여줍니다.



CHAPTER 05 Q 구현 중 겪은 문제점과 해결 방안

테이블 설명

- 1. 계좌 조회 시 비밀번호 비교 로직이 제대로 작동하지 않음
- 2. 이체 시 출금만 되고 입금이 안 되는 현상 발생
- 3. 데이터베이스 연결이 안 된 상태에서 기능을 써서 오류 가 자주 발생함

해결 방안

- 1. 비밀번호와 DB의 값을 정확히 비교하도록 쿼리 수정
- 2. 출금과 입금을 하나의 트랜잭션으로 묶어 처리하고 ROLLBACK 추가
- 3. JDBC 드라이버 파일의 경로를 다시 확인하고 올바르게 설정해 문제를 해결함







CHAPTER 06 Q

느낀점

2주 동안 배운 자바 문법을 실제로 적용해보며 클래스 구조와 데이터 흐름을 더 깊이 이해할 수 있었습니다.

처음에는 어려웠지만, 직접 기능을 하나씩 구현하면서 자신감이 생겼고 특히 객체지향적인 구조와 예외 처리에 대한 감이 잡혔습니다.