

< REPORT 1 >

은행 계좌 프로그램

학과	글로벌미디어학부
학번	20241820
이름	김소현

- 실행 스샷

1. 초기 계좌 정보 입력

```
계좌 번호를 입력하세요: 1234-56-7890
계좌 소유자 이름을 입력하세요: 김소현
==== 계좌 정보 ====
계좌번호: 1234-56-7890
계좌 소유주: 김소현
현재 잔액: 0원
=====
```

2. 입금 / 출금 / 잔액조회

```
===== MENU =====
1: 입금
2: 출금
3: 잔액조회
4: 계좌정보 조회
5: 거래내역 조회
6: 종료
=====
원하시는 작업을 선택하세요: 1
입금할 금액을 입력하세요: 15000

===== MENU =====
1: 입금
2: 출금
3: 잔액조회
4: 계좌정보 조회
5: 거래내역 조회
6: 종료
=====
원하시는 작업을 선택하세요: 2
출금할 금액을 입력하세요: 2500

===== MENU =====
1: 입금
2: 출금
3: 잔액조회
4: 계좌정보 조회
5: 거래내역 조회
6: 종료
=====
원하시는 작업을 선택하세요: 3
현재 잔액: 12500원
```

3. 계좌내역 조회 / 거래내역 조회

```
===== MENU =====
1: 입금
2: 출금
3: 잔액조회
4: 계좌정보 조회
5: 거래내역 조회
6: 종료
=====
원하시는 작업을 선택하세요: 4
===== 계좌 정보 =====
계좌번호: 1234-56-7890
계좌 소유주: 김소현
현재 잔액: 12500원
=====

===== MENU =====
1: 입금
2: 출금
3: 잔액조회
4: 계좌정보 조회
5: 거래내역 조회
6: 종료
=====
원하시는 작업을 선택하세요: 5
15000원이 입금되었습니다.
2500원이 출금되었습니다.
```

4. 이자

```
원하시는 작업을 선택하세요: 1
입금할 금액을 입력하세요: 5000

===== MENU =====
1: 입금
2: 출금
3: 잔액조회
4: 계좌정보 조회
5: 거래내역 조회
6: 종료
=====
원하시는 작업을 선택하세요: 1
입금할 금액을 입력하세요: 7000

===== MENU =====
1: 입금
2: 출금
3: 잔액조회
4: 계좌정보 조회
5: 거래내역 조회
6: 종료
=====
원하시는 작업을 선택하세요: 2
출금할 금액을 입력하세요: 2000

===== MENU =====
1: 입금
2: 출금
3: 잔액조회
4: 계좌정보 조회
5: 거래내역 조회
6: 종료
=====
원하시는 작업을 선택하세요: 3
현재 잔액: 11000원
```

5. 거래내역 조회로 이자 확인 가능

```
===== MENU =====
1: 입금
2: 출금
3: 잔액조회
4: 계좌정보 조회
5: 거래내역 조회
6: 종료
=====
원하시는 작업을 선택하세요: 5
5000원이 입금되었습니다.
7000원이 입금되었습니다.
2000원이 출금되었습니다.
이자 지급: 1000
```

6. 잔액보다 많은 출금 시 오류메시지

```
===== MENU =====
1: 입금
2: 출금
3: 잔액조회
4: 계좌정보 조회
5: 거래내역 조회
6: 종료
=====
원하시는 작업을 선택하세요: 3
현재 잔액: 11000원

===== MENU =====
1: 입금
2: 출금
3: 잔액조회
4: 계좌정보 조회
5: 거래내역 조회
6: 종료
=====
원하시는 작업을 선택하세요: 2
출금할 금액을 입력하세요: 12000
잔액이 부족합니다.
```

7. 잔액 0원일 때 출금 시 오류메시지

```
===== MENU =====
1: 입금
2: 출금
3: 잔액조회
4: 계좌정보 조회
5: 거래내역 조회
6: 종료
=====
원하시는 작업을 선택하세요: 3
현재 잔액: 0원

===== MENU =====
1: 입금
2: 출금
3: 잔액조회
4: 계좌정보 조회
5: 거래내역 조회
6: 종료
=====
원하시는 작업을 선택하세요: 2
출금할 금액을 입력하세요: 1000
잔액이 부족합니다.
```

- 주요 코드 설명

이 코드는 은행 계좌 관리 프로그램을 구현한 것입니다. Calculator 클래스를 통해 계좌 정보를 관리하고 사용자와 상호작용할 수 있는 기능을 제공하는 코드를 작성하였습니다. 전체적으로 흐름을 보면, 프로그램 시작 시 사용자로부터 계좌정보를 입력받고 사용자는 여러 작업(입금, 출금 등)을 선택하여 계좌를 관리할 수 있도록 하였습니다. 거래내역은 벡터를 통해 기록되며, 이자는 거래 횟수에 따라 자동으로 계산됩니다.

1. <헤더파일> BankAccount.h

헤더파일에서는 클래스를 정의해 주었습니다. Calculator 클래스에 여러 멤버변수와 멤버함수를 포함하여 계좌정보를 관리하도록 하였습니다.

멤버함수로 init, deposit, withdraw, displayBalance, displayAccountInfo, displayAccountHistory, payInterest를 지정해 주었습니다. 각 기능은 다음과 같습니다.

- init: 계좌정보 및 멤버변수 초기화
- deposit: 입금 기능 구현
- withdraw: 출금 기능 구현
- displayBalance: 잔액 출력
- displayAccountInfo: 계좌 정보 출력
- displayAccountHistory: 거래 내역 출력
- payInterest: 일정 거래 횟수(3회)마다 이자 0.1% 지급

멤버변수로 accountNumber, accountName, balance, transactionHistory, transactionCount를 지정

해 주었습니다. 각 기능은 다음과 같습니다.

- accountNumber: 계좌 번호
- accountName: 계좌 소유주명
- balance: 계좌 잔액
- transactionHistory: 거래 내역을 저장하는 벡터
- transactionCount: 거래 횟수

2. <소스파일> BankAccount.cpp

BankAccount.cpp 소스파일에서는 각 함수를 구현해 주었습니다.

- init: 계좌정보 및 멤버변수를 초기화하는 기능 구현
- deposit: 입력한 금액을 잔액에 추가하고 거래내역에 기록하는 기능 구현
- withdraw: 잔액이 충분할 경우에만 출금하고 거래내역에 기록하는 기능 구현, 잔액 부족 시 잔액 부족 메시지 출력 기능 구현
- displayBalance: 현재 잔액 출력 기능 구현
- displayAccountInfo: 계좌 정보 출력 기능 구현
- displayAccountHistory: 거래 내역 출력 기능 구현
- payInterest: 일정 거래 횟수가 3의 배수일 때 이자 0.1%를 잔액에 추가하는 기능 구현

3. <소스파일> main.cpp

Main.cpp 소스파일에서는 메인함수를 구현해 주었습니다.

1. 사용자로부터 계좌 번호와 소유자 이름을 입력받아 계좌를 초기화하는 기능 구현
2. 계좌 정보를 출력한 후, 사용자가 원하는 작업을 선택할 수 있는 메뉴 제공 기능 구현
3. 각 메뉴 옵션에 따라 입금, 출금, 잔액조회, 계좌정보 조회, 거래내역 조회를 수행하는 기능 구현
4. 종료를 선택할 경우 프로그램이 종료되는 기능 구현