

REPORT 2

은행 계좌 프로그램2

요구사항(1)

- 샘플코드와 테스트 예시를 참고하여 아래의 요구사항들을 충족하는 코드를 완성하세요.
- (1) 클래스 상속
 - 부모 클래스
 - BankAccount
 - 멤버 변수 : 계좌번호, 이름, 잔액 정보
 - 멤버 함수 : 입금, 출금
 - 정적 멤버 변수 : 총 계좌 수 정보, 총 계좌 잔액 정보
 - 정적 멤버 함수 : 총 계좌 수 및 잔액 정보 출력
 - 자식 클래스
 - SavingsAccount (저축용 계좌)
 - 멤버 변수 : 이자율 정보
 - 멤버 함수 : 이자 적용 함수
 - MinusAccount (마이너스 계좌)
 - 멤버 변수 : 마이너스 출금 한도 금액
 - 멤버 함수 : 출금 (초과 인출 가능)

요구사항(2)

- (2) 계좌 간 자금 이체 기능(전역 함수)

- `void transferFunds(BankAccount& from, BankAccount& to, double amount)`
 - from : 출금 계좌
 - to : 입금 계좌
 - amount : 입금액

- (3) 연산자 중복 정의

- << 출력 연산자
 - 계좌 정보 출력 : 계좌번호, 소유자, 잔액 정보

샘플 코드

```
// Base class: 은행계좌 클래스
class BankAccount {
protected:
    static int totalAccounts;    // 총 계좌 수 (정적 멤버)
    static double totalBalance; // 총 잔액 (정적 멤버)
    string accountNumber;       // 계좌번호
    string ownerName;           // 소유주
    double balance;             // 계좌잔액

public:
    BankAccount(string accNum, string owner, double bal) {
        ...
    }
    ~BankAccount() { ... }

    // 입금 및 출금 함수
    void deposit(double amount) { ... }
    void withdraw(double amount) { ... }
    // 총 계좌수 및 총 잔액정보 출력
    static void displayBankStats() { ... }
    ...
};
```

```
// 저축용 계좌 클래스
class SavingsAccount : public BankAccount {
    double interestRate; // 저축 이자율
public:
    SavingsAccount( string accNum, string owner,
                    double bal, double rate )
        ...
};

// 마이너스 계좌 클래스
class MinusAccount : public BankAccount {
    double minusLimit; // 출금 한도 금액
public:
    MinusAccount( string accNum, string owner,
                  double bal, double limit )
        ...
};

// 계좌 간 자금 이체 함수(전역 함수)
void transferFunds( ... ) {
    ...
}
```

테스트 예시

Microsoft Visual Studio 디버그 콘솔

초기 계좌 통계:
총 계좌 수: 2, 총 잔액: 1500

계좌 정보:
계좌 번호: SA001
소유자: 홍길동
잔액: 1575
계좌 번호: MA001
소유자: 이순신
잔액: -100

최종 계좌 통계:
총 계좌 수: 2, 총 잔액: 1475

```
int main() {  
    SavingsAccount acc1("SA001", "홍길동", 1000, 0.05); // 계좌(저축) 생성  
    MinusAccount acc2("MA001", "이순신", 500, 200); // 계좌(마이너스) 생성  
                                                    // 출금 한도금액  
    // 초기 은행 통계 출력  
    cout << "초기 계좌 통계:\n";  
    BankAccount::displayBankStats();  
  
    // 입금 및 출금 테스트  
    acc1.deposit(500);  
    acc2.withdraw(600); // 초과 인출 허용  
  
    // 이자 적용  
    acc1.applyInterest();  
  
    // 계좌 정보 출력  
    cout << "\n계좌 정보:\n";  
    cout << acc1;  
    cout << acc2;  
  
    // 자금 이체  
    transferFunds(acc1, acc2, 200);  
  
    // 최종 은행 통계 출력  
    cout << "\n최종 계좌 통계:\n";  
    BankAccount::displayBankStats();  
  
    return 0;  
}
```

제출 방법 및 기한

- 제출 일자 : ~ 2024년 12월 7일(토) 자정까지
- 제출 방법
 - 수업사이트(LMS)의 " 평가항목 " 메뉴의 과제 항목 화면에서 제출함
 - 하나의 PDF 문서 파일로 저장하여 LMS 사이트에 업로드
 - 제출 항목 : **문서와 소스코드 모두 제출해야 함!!!**
 - 문서
 - 표지 : 제목, 학과, 학번, 이름
 - 실행 스샷
 - 주요 코드에 대한 설명
 - 소스코드 파일