프로젝트 최종 보고서

|  |
| --- |
| 1. 프로그램 개요   * 은행 거래 관리 프로그램 * 회원 로그인, 입출금 및 송금, 대출, 상환, 거래 내역 출력 기능 * 기존 데이터 로드 및 업데이트   2. 전체 논리 구조도  3. 모듈 별 기능   * + 1. App        1. 기능 총괄 – Transaction, Loan, User을 관리함        2. 회원 로그인 – 회원 ID와 비밀번호를 확인하여 개인 정보를 관리함.        3. 데이터 파일 – 데이터를 열거나 저장함.        4. 메뉴 출력 – 사용자가 원하는 기능을 선택할 수 있도록 출력함.     2. Transaction        1. 거래 내역 관리 – 보낸 사람 ID, 받는 사람 ID, 금액 등을 데이터로 남겨 저장.        2. 거래 정보 출력 – 사용자의 거래 정보를 화면에 출력함.     3. User        1. 사용자 정보 출력 – ID, 계좌번호 등의 정보를 출력함.        2. 사용자의 거래 내역 캐싱 – 데이터를 일시적으로 캐시에 불러오게 함.        3. 사용자 정보 입력 및 반환 – 입, 출금이나 송금, 대출 등 사용자 정보에서 변한 내용을 갱신함.     4. Loan        1. 대출 내역 관리 – 대출한 사용자의 ID, 대출 금액, 남은 상환 기간을 데이터로 남겨 저장.        2. 대출 정보 출력 – 사용자의 대출 정보를 화면에 출력함.   4. 처리 흐름도  기존파일 로드/업데이트  로그인/로그아웃  대출  송금  입금/출금  대출 상환  메뉴  거래내역조회  암호 입력  금액 입력  계좌 입력  금액 입력  금액 입력  반환 기간 입력  남은 금액/반환기간  최근 거래내역 출력  거래 내역 추가 / 갱신  5. 각 모듈의 함수    a. Application();  /\*======================================= Application ==============================================  Application(); // 생성자, 파일 로드 및 변수 초기화  ~Application();//소멸자  void printMenu(); // 기본 메뉴 출력  void print\_logined\_user\_transanctions(); // 로그인된 유저의 거래 내역 출력  void print\_logined\_user\_loans(); // 로그인된 유저의 대출 출력  void print\_deposit\_withdraw\_menu(); // 입,출금 메뉴 출력  void print\_send\_money\_menu(); // 송금 메뉴 출력  void print\_loan\_menu(); // 대출 메뉴 출력  void print\_repayment\_menu(); // 상환 메뉴 출력  User\* login(int user\_id, string Password); // 사용자 로그인  void logout(); // 사용자 로그아웃  bool sendMoney(int \_toAccount, int \_amount); // 파라미터로 받은 계좌번호로 금액을 송금  bool deposit(int \_amount); // 파라미터로 받은 금액만큼 입금  bool withdraw(int \_amount); // 파라미터로 받은 금액만큼 출금  bool loan\_money(int \_amount, int \_repayment\_priod\_month); // 금액 만큼 대출, 상환 기간 설정  bool repayment(); // 대출 시 설정한 금액/반환기간에 맞게 대출금 반환  void load\_user\_file(); // 사용자 파일을 불러옴  void load\_transaction\_log\_file(); // 거래 내역 파일을 불러옴  void load\_loan\_list\_file(); // 대출 리스트 파일을 불러옴  void saveFiles(); // 현재까지의 모든 데이터를 파일로 저장  ==================================================================================================\*/  b. User  /\*======================================= User==============================================  User();//생성자  ~User();//소멸자  int getId(); // 아이디 리턴  string getPW(); // 비밀번호 리턴  string getName(); // 이름 리턴  int getAccoutNumber(); // 계좌번호 리턴  int getBalance(); // 잔금 리턴  int getCreditRating(); // 신용등급 리턴  int getCacheSize(); // 현재 캐시 사이즈 리턴  void setBalance(int \_balance); // 잔금 설정  void setCreditRating(int \_creditRating); // 신용등급 설정  void setCache(transaction\* trans); // 현재 유저의 캐시에 trans 배열의 기록을 저장  void resetCache(){ cacheSize = 0; }; // 캐시 사이즈를 초기화  transaction\*\* getTrasCache(); // Application의 Transaction DB에서 ID가 일치하는 거래내역을 가져옴  void setUser(int \_id, string \_password, string \_name, int \_accountNumber, int \_balance, int \_creditRating); // 사용자 정보 입력  void printUserInfo(); // 사용자 정보 출력    =============================================================================================\*/  c. loan;  /\*======================================= loan==============================================  loan();//생성자  ~loan();//소멸자  void setLoan(int \_id, int \_amount, int \_repayment\_priod\_month); // 대출 정보 입력  int getUserId(); // 사용자 아이디 반환  int getAmount(); // 대출 잔금 반환  int GetRepayment\_Period\_Month(); // 잔여 상환 기간 반환  void setAmount(int \_amount); // 대출금 변경  void setRepayment\_Period\_Month(int \_period); // 잔여 상환 기간 변경  ==================================================================================================\*/0  d. transaction  /\*======================================= transaction==============================================  transaction();//생성자  ~transaction();//소멸자  int getFromId(); // 보낸 사람 아이디 리턴  int getToId(); // 받은 사람 아이디 리턴  int getAmount(); // 금액 리턴  void setTransation(int \_fromId, int \_toId, int \_amount); // 거래 정보 입력  void print(); // 거래 정보 출력  ==================================================================================================\*/  6. 설계 원칙  설계 요구 조건에 따라 충실히 설계한다.   * 로그인으로 본인 확인 후 이용 가능하며, 이용 가능한 메뉴는 입, 출금 / 송금 / 대출 / 대출금 반환 / 거래내역 조회로 한다. * 프로그램 시작 시, user.txt, transaction.txt, loan.txt로부터 데이터를 불러와 각각 User와 transaction, loan 객체 배열에 저장시킨다. * App은 User DB, transaction DB, loan DB를 가지고 있으며, User와 transaction, loan들을 총괄한다. 또한 관리에 필요한 함수를 호출한다 * User는 아이디, 비밀번호, 사용자 이름, 계좌번호, 잔액, 신용등급, 거래내역 캐시를 가지고 있다. * 프로그램 종료 시에 사용자의 거래내역을 계산하여 잔액과 신용등급을 갱신하고, User DB를 저장한다. * 예외 처리는 다음과 같은 경우에 사용한다. * 잔액이 부족하여 대출금 상환이 불가능한 상황, 대신 신용등급을 강등 * 잔액이 부족하여 송금 및 출금이 불가한 경우 * 잘못된 금액을 넣었을 경우(음수, 문자 등) * 메뉴 선택 시 메뉴의 리스트 이외의 값을 넣었을 경우 * 잘못된 아이디 혹은 비밀번호로 로그인 시도 시   대출을 상환에 한번 성공할 때마다 신용등급이 1씩 올라가고, 잔액부족으로 실패할 때 1씩 떨어진다. 대출 가능한 금액은 신용등급에 따라 상한선이 정해진다. 신용등급이 1이면 1000만원까지 가능하며, 1씩 하락할 때마다 100만원씩 한도가 줄어들며, 신용등급이 10이면 대출이 불가하다. 단, 이미 대출한 내역이 있으면, 모두 상환하기 전까지 대출이 불가능하다.  대출금 상환 시 한번 반환할 때 마다 한 달이 지난 것으로 가정하고, 남은 상환금액을 남은 상환 기간만큼 나눈 몫을 상환한다. 상환 마지막 달은 남은 금액을 모두 반환한다. 상환한 내용은 거래내역에서 출금 형식으로 확인할 수 있다.  7. 개발도구  Microsoft Visual Studio 2013 C++  8. 사용 매뉴얼  a. 시작 메뉴     1. 로그인 : 입력 데이터와 저장된 ID, 비밀번호를 확인하여 개인 계정에 접속, 개인 정보와 메뉴 출력 (예외 처리 : 일치하는 ID가 없을 경우, 비밀번호가 일치하지 않을 경우, 이미 로그인 된 상태에서 재 로그인 할 경우, 지정된 숫자 외 다른 문자를 입력 받았을 경우) 2. 로그아웃 : 현재 로그인 된 계정의 접속을 종료하고 로그인 메뉴로 이동 3. 종료 : 현재까지의 모든 데이터를 저장   b. 입/출금     1. 입금 : 현재 로그인 된 사용자의 잔액에 입력 받은 금액을 추가되며 최근 거래 내역 리스트에 추가. (예외 처리 : 지정된 숫자 외 다른 문자를 입력 받았을 경우, 입금 금액이 음수인 경우) 2. 출금 : 현재 로그인 된 사용자의 잔액을 입력 받은 금액만큼 차감. 최근 거래 내역 리스트에 추가. (예외 처리 : 지정된 숫자 외 다른 문자를 입력 받았을 경우, 출금 금액이 음수인 경우)   c. 송금     1. 송금 : 받는 사람의 계좌번호, 금액을 입력 받고 돈을 송금함. 최근 거래 내역 리스트에 추가. 2. (예외처리 : 송금 금액이 음수인 경우, 자기 자신에게 송금하는 경우, 송금 금액이 잔액을 넘어선 경우, 지정된 숫자 외 다른 문자를 입력 받았을 경우)   d. 대출     1. 대출 : 대출 금액과 반환 기간을 입력 받은 후, 잔액을 추가. 최근 거래 내역 리스트에 추가. 신용 등급이 1일경우 1000만 원까지, 2일경우 900만 원, … , 9일 경우 200만 원까지 가능하며 10일 경우 대출이 불가능함. 아직 잔여 대출금이 있다면 추가로 대출은 불가능함. 2. (예외처리 : 대출금을 다 갚지 않았을 경우, 신용 등급을 넘어선 대출금액을 입력했을 경우, 올바르지 않은 대출금액 또는 상환기간이 입력된 경우, 신용 등급이 10인 경우)  .   e. 대출금 상환   1. 대출금 상환 : 대출 시 설정한 상환 기간에 따라 대출금을 상환 함. 한 번 상환 시 마다 1개월이 지난 것으로 간주함. 상환 성공 시 신용 등급이 상승함. 잔액 부족으로 상환 실패 시 신용 등급이 하락. (예외 처리 : 잔액 부족으로 상환 실패할 경우, 대출 내역이 존재하지 않을 경우 등)   f. 거래내역 출력     1. 거래내역 출력 : 모든 유저의 활동들은 거래내역 리스트에 저장되고, 그 중 현재 로그인 한 유저의 최근 5개 거래 내역을 출력. 송금/출금/대출금 상환 등 출금인과 입금인이 없을 경우에는 -1로 표기함. |
|  |