하위 설계서

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | 1. 목적 |  * 고급객체지향 프로그래밍의 실습 과제이다. 클래스, 포인터 등을 사용한 프로그래밍을 통하여 실력을 향상시킨다. * 실전 프로그래밍에서의 예외 처리, 다양한 선택변수 등을 접하고 대비하는 능력을 향상시킨다.  |  | | --- | | 1. 사용자 Targeting |   입출금 및 대출과 같은 은행을 이용하고자 하는 고객을 대상으로 하며, 직관적인 인터페이스로 누구나 쉽게 은행 거래 관리 프로그램을 이용할 수 있어야 한다.   |  | | --- | | 1. 설계 원칙 |  * 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.   |  | | --- | | 1. 개발 도구 |   Microsoft Visual Studio C++ 2013   |  | | --- | | 1. 적용 범위 |   장소적 범위 : 은행 거래 관리 시스템을 설치할 수 있는 모든 장소  프로그램적 범위 : ATM 프로그램, 온라인 뱅킹 프로그램 등   |  | | --- | | 1. 시나리오 흐름도 |   시나리오 흐름도1 (로그인)    시나리오 흐름도2 (입/출금, 송금)    시나리오 흐름도3 (대출/상환/거래내역조회)     |  | | --- | | 1. 인터페이스 정의 |  * 문장은 사용자가 해당 기능에 대한 사용법을 명확히 알 수 있도록 명시한다. * 먼저 로그인 메뉴를 출력한 뒤, 로그인이 성공적으로 이루어졌으면 선택 메뉴를 출력한다. * 글자가 지저분하지 않도록 띄어쓰기와 줄 바꿈을 해준다. * 입력은 키보드로만 받는다. * 한국어와 영어 외의 언어는 사용하지 않는다.  |  | | --- | | 1. 헤더와 cpp, 클래스정의 |  * 클래스 선언은 헤더파일을 만들고 메인 cpp에 불러온다. * 헤더에서 선언한 클래스는 헤더와 같은 이름의 cpp에 구현한다.  |  |  | | --- | --- | | **헤더와 cpp** | | | 기능 | 이름 | | 클래스 선언 | Application.h  Loan.h  Transaction.h  User.h | | 클래스 구현 | Application.cpp  Loan.cpp  Transaction.h  User.h | | 메인 구현 | Main.cpp |  |  |  | | --- | --- | | **클래스** | | | 클래스 이름 | 기능 | | Application | Transaction, Loan, User 관리  회원 로그인  데이터 파일 입출력  메뉴 출력 | | Loan | 대출 내역 관리 및 출력 | | Transaction | 거래 내역 관리 및 출력 | | User | 사용자 정보 입력, 갱신 및 출력  사용자 거래 내역 캐싱 |   ▣ 클래스 별 요소와 기능 ▣   |  | | --- | | **Application** | | Private   |  |  | | --- | --- | | 변수이름 | 기능 | | User users[20] | 유저의 개인 정보 저장 배열  거래 내역 캐시를 불러와 저장 | | Transaction transactions[100] | 거래 내역과 관련된 정보 배열 | | Loan loans[100] | 대출과 관련된 정보 저장 배열 | | User\* loginedUser | 로그인 된 유저의 정보를 불러옴 |  |  |  | | --- | --- | | 함수(메소드) 이름 | 기능 | | printMenu | 메뉴 출력 | | Print\_logined\_user\_transactions | 로그인 된 사용자의 거래 내역 출력 | | Print\_logined\_user\_loans | 로그인 된 사용자의 대출 | | Print\_deposit\_withdraw\_menu | 입출금 메뉴 출력 | | Print\_send\_money\_menu | 송금 메뉴 출력 | | Print\_loan\_menu | 대출 메뉴 출력 | | Print\_repayment\_menu | 상환 메뉴 출력 | | Login | 사용자 로그인 | | Logout | 사용자 로그아웃 | | User\* getUserbyId | 유저 데이터를 불러옴 | | Sendmoney | 송금 | | Deposit | 입금 | | Withdraw | 출금 | | Loan\_money | 대출 | | repayment | 대출금 상환 | | Load\_user\_file | 사용자 파일 로드 | | Load\_transaciton\_log\_file | 거리 내역 파일 로드 | | Load\_loan\_list\_file | 대출 리스트 파일 로드 | | saveFiles | 파일 저장 |   Public   |  |  | | --- | --- | | 함수(메소드) 이름 | 기능 | | run | 시작 화면 출력 | |  |  | | --- | | **Loan** | | Private   |  |  | | --- | --- | | 변수이름 | 기능 | | userId | 사용자 ID | | amount | 대출 금액 | | repayment\_priod\_month | 대출 상환 기간 |   public   |  |  | | --- | --- | | 함수(메소드) 이름 | 기능 | | setLoan | 대출 정보 입력 | | Print | 대출 정보 출력 | | Loan\_money | 대출 받은 만큼 현재 유저의 잔액을 갱신하고, 거래 내역을 추가함 | | repayment | 상환 금액 만큼 현재 유저의 잔액을 갱신하고, 거래 내역을 추가함 | |  |  | | --- | | **Transaction** | | Private   |  |  | | --- | --- | | 변수이름 | 기능 | | fromId | 보내는 사람 ID | | ToId | 받는 사람 ID | | amount | 금액 |   public   |  |  | | --- | --- | | 함수(메소드) 이름 | 기능 | | setTransaction | 거래 정보 입력 | | Print | 거래 정보 출력 | |  |  | | --- | | **User** | | Private   |  |  | | --- | --- | | 변수이름 | 기능 | | Id | ID | | Password | 비밀번호 | | Name | 이름 | | accountNumber | 계좌번호 | | Balance | 잔고 | | creditRating | 신용 등급 | | chacheSize | 캐싱 될 수 있는 거래 내역의 개수를 저장 | | Transaction\* TransCache[5] | 5개의 거래 내역을 캐싱할 수 있는 배열 |   public   |  |  | | --- | --- | | 함수(메소드) 이름 | 기능 | | getTrasCache | App의 Transaction DB에서 ID가 일치하는 거래 내역을 가져옴 | | setUser | 사용자 정보 입력 | | printUserInfo | 사용자 정보 출력 | |  |  | | --- | | 1. Main의 변수와 cpp정의 |   Application : Application 클래스 선언   |  | | --- | | 1. 프로젝트 추진 일정 |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | **추진일정** | | | | | | | | | | | | **1주** | | **2주** | | **3주** | | **4주** | | **5주** | | | | **요구사항 분석** |  |  |  |  |  |  |  |  |  |  |  | | **헤더,  cpp작성** |  |  |  |  |  |  |  |  |  |  |  | | **main작성** |  |  |  |  |  |  |  |  |  |  |  | | **1차 완성** |  |  |  |  |  |  |  |  |  |  |  | | **옵션 추가** |  |  |  |  |  |  |  |  |  |  |  | | **버그 수정** |  |  |  |  |  |  |  |  |  |  |  | | **검토 및  제출** |  |  |  |  |  |  |  |  |  |  |  | |