하위 설계서

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | 1. 목적 |  * 고급객체지향 프로그래밍의 실습 과제이다. 클래스, 상속 등을 사용한 프로그래밍을 통하여 실력을 향상시킨다. * 실전 프로그래밍에서의 예외 처리, 다양한 선택변수 등을 접하고 대비하는 능력을 향상시킨다.  |  | | --- | | 1. 사용자 Targeting |  * 방 추천 프로그램을 이용하고자 하는 고객 및 관리자를 대상으로 하며, 직관적인 인터페이스로 누구나 쉽게 방 추천 프로그램을 이용할 수 있어야 한다.  |  | | --- | | 1. 설계 원칙 |  1. 방 추천 프로그램을 구현하기 위한 클래스는 App, User, Seller, Buyer, Room, RoomForRent, RoomForSale 이다. 2. User 클래스를 상속받은 구매자 클래스 Seller와 판매자 클래스 Buyer가 있다. 3. Room 클래스를 상속받은 월세 클래스 RoomForRent와 매매 클래스 RoomForSale가 있다. 4. App은 User DB, Room DB를 가지고 있다. 5. User는 아이디, 비밀번호, 사용자 타입을 가지고 있다.    * User를 상속받은 Seller는 자신이 등록한 매물 목록을 가지고 있다.    * User를 상속받은 Buyer는 자신의 좌표, 최근 검색 내역, 보낸 구매요청 리스트를 가지고 있다. 6. App는 User와 Room 객체들을 총괄하여 관리하고 필요한 함수를 호출한다. 7. 프로그램 실행 시에 User DB, Room DB를 App 클래스의 각 컨테이너에 load한다. 8. 프로그램 종료 시에 갱신된 정보들을 파일에 다시 쓴다. 9. 구매자 기능(기본 정보, 검색 필터 설정, 방 검색, 구매 신청, 최근 검색 내역) 과 판매자 기능(기본 정보, 방 등록, 방 삭제)을 구현한다.  |  | | --- | | 1. 개발 도구 |   Microsoft Visual Studio C++ 2013   |  | | --- | | 1. 적용 범위 |  * 장소적 범위 : 방 추천 프로그램이 사용될 수 있는 장소(부동산 매매 등) * 프로그램적 범위 : 스마트 폰 어플리케이션  |  | | --- | | 1. 시나리오 흐름도 |   시나리오 흐름도     |  | | --- | | 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 | App.h  Buyer.h  Room.h  RoomForRent.h  RoomForSale.h  Seller.h  User.h | | 클래스 구현 | App.cpp  Buyer.cpp  Room.cpp  RoomForRent.cpp  RoomForSale.cpp  Seller.cpp  User.cpp | | 메인 구현 | Main.cpp |  |  |  | | --- | --- | | **클래스** | | | 클래스 이름 | 기능 | | App | Room, User 클래스 관리  회원 로그인  데이터 파일 입출력 및 메뉴 출력  함수 호출  기타 모든 기능 총괄 | | User | 프로그램 사용자 기본 정보 관리 | | Seller | 등록한 방 정보 관리 | | Buyer | 좌표, 최근 검색 내역, 구매 요청 내역 관리 | | Room | 방의 기본 정보 관리 | | RoomForRent | 보증금, 월세 관리 | | RoomForSale | 매매가 관리 |   ▣ 클래스 별 요소와 기능 ▣   |  | | --- | | **App** | | #pragma once  #include "Seller.h"  #include "Buyer.h"  #include "RoomForRent.h"  #include "RoomForSale.h"  #include <vector>  using namespace std;  class App  {  public:  App(); //기본 생성자  ~App(); //소멸자  void run(); //프로그램 시행  private:  vector<User\*> users; //유저 벡터  vector<Room\*> rooms; //방 벡터  User\* loginedUser; //로그인 유저  vector<Room\*> reusltOfFilter1, resultOfFilter2; //조건1, 2에 대한 필터링을 수행한 임시 검색 결과  int searchAreaRadius; //사용자 중심으로 검색할 범위(거리)  bool isFilter1Skipped, istFilter2Skipped, isFilter3Skipped; //필터링을 수행할 것인지 저장하는 변수들  int minDeposit, maxDeposit; //필터링에 사용될 보증금의 범위를 저장하는 변수들  int minMonthlyPay, maxMonthlyPay; //필터링에 사용될 월세의 범위를 저장하는 변수들  int minCost, maxCost; //필터링에 사용될 매매가의 범위를 저장하는 변수들    vector<Room\*> getFilter1Result(); // 필터1 반환 함수  vector<Room\*> getFilter2Result(); // 필터2 반환 함수  vector<Room\*> getFilter3Result(); // 필터3 반환 함수  void loadUserFile(); //유저 데이터 로드  void loadRoomFile(); //방 데이터 로드  void saveFiles(); //파일 저장  void run(); //프로그램 시행  void printMenu(); //메뉴 출력  bool login(string userID, string password); //로그인  bool logout(); //로그아웃    void printRecentlySearcedRoomsOfLoginedUser(); //최근 검색 내역 출력  Room\* getRoomByID(int \_roomID); //ID를 이용해 Room 반환  bool printFilterConditions(); //현재 필터들의 설정을 출력  void searchRooms(); //방 검색  void registerNewRoom(); // 방 등록  void deleteRoom(); //방 삭제  }; |  |  | | --- | | **User** | | #pragma once  #include <string>  using namespace std;  class User  {  public:  User(); //기본 생성자  ~User(); //소멸자  virtual void printUserInfo() = 0; //정보를 출력하는 순수 가상함수  private:  int userType; //유저 타입  string id; //아이디  string password; //패스워드    }; |  |  | | --- | | **Buyer** | | #pragma once  #include "User.h"  #include <list>  class Buyer :  public User  {  public:  Buyer(); //기본 생성자  ~Buyer(); //소멸자  bool addRoomIDToRecentlySearched(int \_roomID); //최근 검색 내역에 추가  bool addRoomIDToRequestToBuy(int \_roomID); //구매 요청 내역에 추가  virtual void printUserInfo(); //구매자 정보 출력  private:  int xCoord, yCoord; //구매자의 좌표  list<int> recentlySearched; //최근 검색 내역 리스트  list<int> roomsRequestToBuy; //구매 요청 내역 리스트    }; |  |  | | --- | | **Seller** | | #pragma once  #include "User.h"  #include <list>  class Seller :  public User  {  public:  Seller(); // 기본 생성자  ~Seller(); // 소멸자  void addRoomIDToRegisteredRoomList(int \_roomID); //iD를 이용해 방 추가  virtual void printUserInfo(); //판매자 정보 출력  private:  list<int> registerdRoomList; //등록한 방 리스트  }; |  |  | | --- | | **Room** | | #pragma once  #include <list>  #include <string>  using namespace std;  class Room  {  public:  Room(); // 기본 생성자  ~Room(); // 소멸자  bool addUserInRequestToBuyList(string \_userID); // 유저ID를 이용해 구매 요청 리스트에 추가  virtual void printRoomInfo() = 0; // 정보를 출력하는 순수 가상 함수  private:  int roomType; // 방 타입  int roomID; // 방 ID  int sellerID; // 판매자 ID  string roomName; // 방 이름  int xCoord, yCoord; // 좌표  list<string> usersRequestToBuyList; // 구매 요청한 유저 ID 리스트    }; |  |  | | --- | | **RoomForRent** | | #pragma once  #include "Room.h"  class RoomForRent :  public Room  {  public:  RoomForRent(); // 기본 생성자  ~RoomForRent(); // 소멸자  virtual void printRoomInfo(); // 방 정보 출력  private:  int deposit; // 보증금  int monthlyPay; // 월세    }; |  |  | | --- | | **RoomForSale** | | #pragma once  #include "Room.h"  class RoomForSale :  public Room  {  public:  RoomForSale(); // 기본 생성자  ~RoomForSale(); // 소멸자  virtual void printRoomInfo(); // 방 정보 출력  private:  int cost; // 매매가    }; |  |  | | --- | | 1. Main의 변수와 cpp정의 |   App : App 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** |  |  |  |  |  |  |  |  |  |  |  | |