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 |  * 우편물 배송 관리 시스템을 이용하고자 하는 고객 및 관리자를 대상으로 하며, 직관적인 인터페이스로 누구나 쉽게 우편물 배송 관리 시스템을 이용할 수 있어야 한다.  |  | | --- | | 1. 설계 원칙 |  1. User 클래스를 상속받은 사용자 클래스 Client와 우편물 관리자 클래스 Manager가 있다. 2. Mail 클래스를 상속받은 편지 클래스 Letter와 소포 클래스 Package가 있다. 3. App은 User DB, Router DB, Mail Center DB를 가지고 있다.  - User.txt 파일에는 User를 상속받은 Client와 Manager가 섞여 저장되어 있으며 공통적으로 부모인 User가 가지고 있는 userType, ID, pw를 가지고 있다  - MailCenterID.txt 파일에는 전국의 우편집중국과 각 우편집중국의 ID 필드가 있다.  - Router.txt는 우편집중국들 간에 연결정보를 가지고 있다. 4. User는 아이디, 비밀번호, 사용자 타입을 가지고 있다. 5. User를 상속받은 Client는 이름, 주소, 보낸 우편물 목록, 받은 우편물 목록을 가지고 있다. 6. User를 상속받은 Manager는 추가적인 맴버변수는 가지고 있지 않다. 7. Client와 Manager는 User클래스의 순수가상함수 send( )와 receive( ) 함수를 overridding하고 있어 서로 다른 기능을 수행한다. 8. App는 User와 MailCenter 객체들을 총괄하여 관리하고 필요한 함수를 호출한다. 9. 프로그램 실행 시에 User DB, Router DB, Mail Center ID DB를 App 클래스의 각 컨테이너에 로드한다. 10. 프로그램 종료 시에 갱신된 정보들을 파일에 다시 쓰지 않는다. 이번 프로젝트는 난이도를 고려하여 모든 우편물 송수신을 메모리에서만 동작하는 것으로 고려하고, 파일 입출력을 하지 않는다. 단, 앞서 언급한 우편물 관리에 필요한 모든 DB들은 프로그램 시작 시 읽어와야 한다.  |  | | --- | | 1. 개발 도구 |   Microsoft Visual Studio C++ 2013   |  | | --- | | 1. 적용 범위 |   장소적 범위 : 우편물 배송 관리 시스템을 설치할 수 있는 장소, 우체국 등  프로그램적 범위 : 온라인 우편 배송 프로그램 등   |  | | --- | | 1. 시나리오 흐름도 |   시나리오 흐름도1   |  | | --- | | 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  Mail.h  MailCenter.h  MailDelivery.h  User.h | | 클래스 구현 | App.cpp  Mail.cpp  MailCenter.cpp  MailDelivery.cpp  User.cpp | | 메인 구현 | Main.cpp |  |  |  | | --- | --- | | **클래스** | | | 클래스 이름 | 기능 | | App | Mail, MailCenter, MailDelivery, User 관리  회원 로그인  데이터 파일 입출력 및 메뉴 출력  함수 호출 | | Mail | 편지 및 소포 정보 | | MailCenter | 우편집중국 정보 | | MailDelivery | 고객 및 관리자의 우편물 전송/수신 | | User | 고객 및 관리자 정보 |   ▣ 클래스 별 요소와 기능 ▣   |  | | --- | | **App** | | class App  {  private:  User\* users[36]; //전체 고객 배열  MailCenter mailCenters[12]; //전체 우편집중국 배열  int\*\* router; //현재 우편 집중국이 인근 우편집중국과 연결되어 있는지를 확인  User\* loginedUser; //현재 로그인한 유저를 가리키는 포인터 변수  private:  bool load\_files(); //데이터 파일 로드  bool login(); //로그인 함수  bool logout(); //로그아웃 함수  void saveFiles(); //데이터 파일 저장  void printClientMenu(); //고객 메뉴 출력  void printManagerMenu(); //관리자 메뉴 출력  public:  App(); //파일 로드 및 변수 초기화  ~App(); //소멸자  void run(); //시작 화면 출력  }; |  |  | | --- | | **Mail** | | class Mail  {  private:  string senderId; //보낸 사람 ID  string recieverId; //받는 사람 ID  string senderAddr; //보낸 사람 주소  string recieverAddr; //받는 사람 주소  int mailType; //우편물 종류  int cost; //배송 비용  virtual int calculateCost(int distance) = 0; //배송 비용을 계산하는 순수 가상 함수  public:  Mail();  ~Mail();  };  class Letter : public Mail // Mail로부터 상속받은 Letter 클래스  {  public:  virtual int calculateCost(int distance); // 거리에 따른 배송 비용 계산  public:  Letter();  ~Letter();  };  class Package : public Mail // Mail로부터 상속받은 Package 클래스  {  public:  virtual int calculateCost(int distance); // 거리와 무게에 따른 배송 비용 계산  Package();  ~Package();  private:  int weight; // 소포 무게  }; |  |  | | --- | | **MailCenter** | | class MailCenter  {  private:  int id; //우편집중국 ID  string cityName; //도시 이름  int neighborCnt; //인접한 우편집중국 개수  int\* neighborIds; //인접한 우편집중국 아이디  Mail\* recievedMailList[10]; //받은 우편물 리스트  Mail\* localMailList[10]; //내부로 발송될 우편물 리스트  Mail\*\*\* externalMailList; //외부로 발송될 우편물 리스트  public:  MailCenter();  ~MailCenter();  }; |  |  | | --- | | **MailDelivery** | | class mailDeilivery  {  private:  User\*\* pointerOfUsers; //고객을 가리키는 포인터  MailCenter\* pointerOfMailCenters;//우편집중국을 가리키는 포인터  int\*\* pointerOfRouter; //경로를 가리키는 포인터  int getTotalDistance(int sourceId, int destinationId);//출발지에서 목적지까지 총 이동 횟수를 계산하여 반환  int getNextMailCenterId(int currentPos, int destinationId);//현재 우편집중국에서 목적지 우편집까지 가는 경로의 다음 우편집중국 ID를 받아옴  int getMailCenterIdByCityName(string cityName);//도시 이름을 이용해 ID를 받아옴  public:  mailDeilivery();  ~mailDeilivery();    bool clientSendMail(User\* curUser);//고객의 우편물 보내기  bool managerSendMail(string cityName);//관리자의 우편물 보내기  bool managerReceiveMail(string cityName);//관리자의 우편물 받기  void setDBPointers(User\*\* pUser, MailCenter\* pMailCenter, int\*\* pRouter)//App 클래스에 저장된 데이터들의 주소값을 복사해옴  }; |  |  | | --- | | **User** | | class User  {  private:  int userType; // 유저 타입( 0은 일반 고객, 1은 관리자)  string id; //아이디  string password; // 비밀번호  public:  User();  ~User();  virtual bool send(mailDeilivery& md) = 0; // 우편물을 보내는 순수 가상 함수  virtual bool receive(mailDeilivery& md) = 0; // 우편물을 받는 순수 가상 함수  };  class Client : public User // User로부터 상속받은 Client 클래스  {  private:  string name; // 이름  string address; // 주소  Mail\* sentMailList[10]; // 보낸 메일 리스트  Mail\* reciedvedMailList[10]; // 받은 메일 리스트  public:  Client();  ~Client();  virtual bool send(mailDeilivery& md); // 우편물 보내기  virtual bool receive(mailDeilivery& md); // 우편물 받기  };  class Manager : public User // User로부터 상속받은 Manager 클래스  {  public:  Manager();  ~Manager();  virtual bool send(mailDeilivery& md); // 우편물 보내기  virtual bool receive(mailDeilivery& md); //우편물 받기    private:  string currentCityName; // 현재 도시 이름  }; |  |  | | --- | | 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** |  |  |  |  |  |  |  |  |  |  |  | |