C++ 프로그래밍및실습(3)

주소록

진척 보고서 #1

제출일자: 2023.11.26
제출자명: 이규민
제출자학번: 215118

1. 프로젝트 목표

1) 배경 및 필요성

개인이나 조직이 연락처 및 주소록을 효과적으로 관리하고, 업데이트 하는 데 도움을 줄 수 있다.

2) 프로젝트 목표

사용자가 연락처 정보를 추가, 수정, 삭제, 검색할 수 있는 주소록 관리 프로그램을 개발한다. 사용자가 연락처의 이름, 전화번호, 이메일, 주소, 주소 등을 저장하고 해당하는 정보를 검색할 수 있어야 한다.

3) 차별점

기존 주소록 관리 프로그램과의 차별점은 단순히 CRUD를 통한 저장, 읽기, 갱신, 삭제 뿐 아니라, 사용자에게 검색 옵션을 제공하여 특정 연락처를 빠르게 찾을 수 있습니다.

2. 기능 계획

1) 연락처 추가

- 설명 (예: 대상 고객과 유사한 주문내역을 갖는 고객을 찾는 기능)

(1) 필수 필드 입력

- 사용자가 연락처를 추가할 때 필수 정보인 이름과 전화번호를 입력하도록 요구합니다.

(2) 추가정보 입력

- 사용자가 이메일 주소, 주소, 생년월일 등의 추가 정보를 입력할 수 있도록 합니다.

2) 연락처 검색

- 사용자는 이름, 전화번호, 이메일 주소 등의 기준을 사용하여 연락처를 검색할 수 있어야 합니다.

(1) 이름으로 검색

- 사용자는 연락처의 이름으로 검색하여 해당 연락처를 찾을 수 있어야 합니다.

(2) 전화번호로 검색

- 사용자는 연락처의 전화번호로 검색하여 해당 연락처를 찾을 수 있어야 합니다.

3) 연락처 수정

- 사용자는 저장된 연락처의 정보를 수정할 수 있어야 합니다.

(1) 연락처 정보 수정

- 사용자는 연락처의 정보를 수정할 수 있어야 합니다. 변경 가능한 정보는 이름, 전화번호, 이메일 주소, 주소, 생년월일 등이 포함됩니다.

4) 연락처 삭제

- 사용자는 더 이상 필요하지 않는 연락처를 삭제할 수 있어야 합니다.

(1) 연락처 삭제 확인

- 사용자가 연락처를 삭제하기 전에 확인 메시지를 표시하여 실수로 삭제되는 것을 방지합니다.

3. 진척사항

1) 기능 구현

(1) Create - 연락처 추가

- 입출력
    - 사용자로부터 연락처의 정보를 입력받음
- 설명
    - 사용자로부터 이름과 전화번호는 필수 입력으로 받고, 이메일, 주소, 그룹은 선택적으로 받음.
    - 입력받은 정보를 새로운 Contact 객체로 만들어 전역 연락처 목록에 추가.
- 적용된 배운 내용
    - 구조체 사용
    - 벡터 사용
    - 반복문과 조건문 활용
    - 함수 (addContact) 정의와 호출
    - getline 함수를 사용한 문자열 입력
- 코드 스크린샷

```
// 주소록에 연락처 추가
void addContact()
{
    cout << "------------------\n";

    cout << "연락처 정보를 입력하세요:\n";

    Contact newContact;

    // 필수 필드 입력
    cout << "이름 (필수): ";
    getline(cin, newContact.name);

    cout << "전화번호 (필수): ";
    getline(cin, newContact.phoneNumber);

    // 추가 정보 입력
    cout << "이메일: ";
    getline(cin, newContact.email);

    cout << "주소: ";
    getline(cin, newContact.address);

    // 그룹 입력
    cout << "그룹: ";
    getline(cin, newContact.group);

    // 연락처 추가
    contactList.push_back(newContact);

    cout << "연락처를 성공적으로 추가하였습니다!\n";
}
```

(2) Read - 전체 연락처 조회

- 입출력
    - 사용자에게 연락처의 전체 목록을 출력
- 설명
    - 전역 연락처 목록에 있는 각 연락처의 정보를 출력
- 적용된 배운 내용
    - 벡터 사용
    - 반복문과 조건문 활용
    - 함수 (viewAllContacts) 정의와 호출
- 코드 스크린샷

```cpp
// 전체 연락처 출력
void viewAllContacts()
{
    cout << "--------------------\n";

    cout << "==== 전체 연락처 목록 ====\n";
    for (const auto &contact : contactList)
    {
        cout << "이름: " << contact.name << "\n";
        cout << "전화번호: " << contact.phoneNumber << "\n";
        cout << "이메일: " << contact.email << "\n";
        cout << "주소: " << contact.address << "\n";
        cout << "그룹: " << contact.group << "\n"; // 그룹 출력 추가
        cout << "--------------------\n";
    }
}
```

(3) Update - 연락처 수정

- 입출력
    - 사용자에게 수정할 연락처의 이름과 새로운 정보를 입력받음
- 설명
    - 입력받은 이름으로 연락처를 찾고, 찾은 경우에는 해당 연락처의 정보를 사용자에게서 다시 입력받아 업데이트
- 적용된 배운 내용
    - 구조체 사용
    - 벡터 사용
    - 반복문과 조건문 활용
    - 함수 (modifyContact) 정의와 호출
    - getline 함수를 사용한 문자열 입력
- 코드 스크린샷

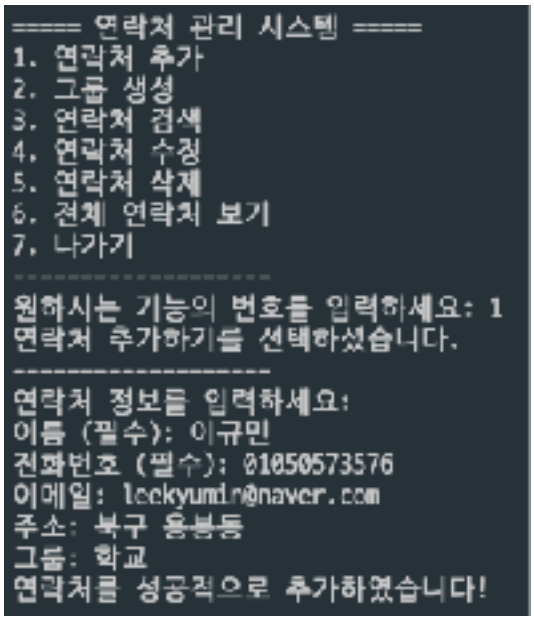
(4) Delete - 연락처 삭제

- 입출력
    - 사용자로부터 삭제할 연락처의 이름을 입력 받음
- 설명
    - 입력받은 이름으로 연락처를 찾고, 찾은 경우에는 해당 연락처를 목록에서 삭제
- 적용된 배운 내용
    - 구조체 사용
    - 벡터 사용
    - 반복문과 조건문 활용
    - 함수 (deleteContact) 정의와 호출
    - getline 함수를 사용한 문자열 입력
- 코드 스크린샷

2) 테스트 결과

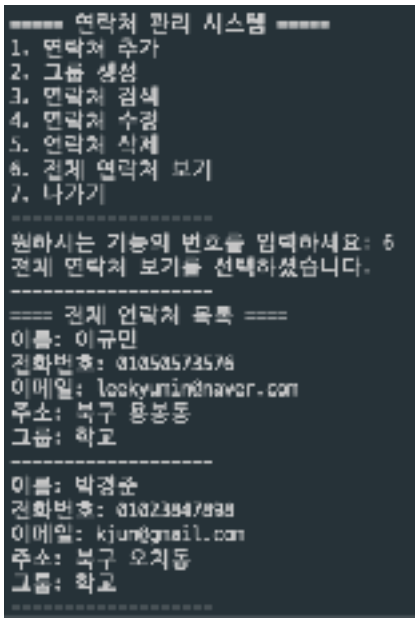
(1) Create - 연락처 추가 테스트

- 사용자로부터 연락처의 정보를 입력받음
    - 입력이 제대로 되는지 테스트합니다
- 테스트 결과 스크린샷



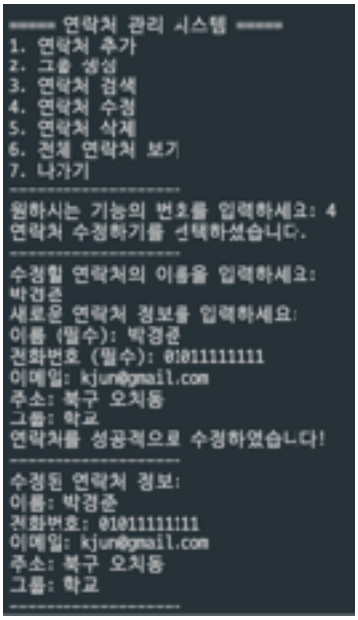
(2) Read - 전체 연락처 조회 테스트

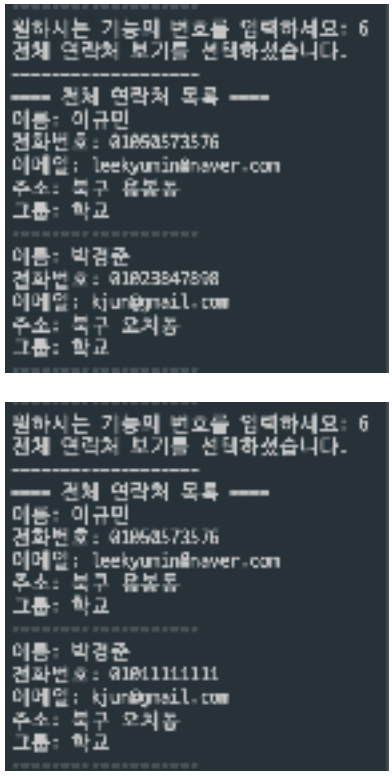
- 전역 연락처 목록에 있는 각 연락처의 정보를 출력
    - Create를 이용해 추가된 모든 연락처가 제대로 출력되는지 테스트합니다
- 테스트 결과 스크린샷



(3) Update - 연락처 수정 테스트

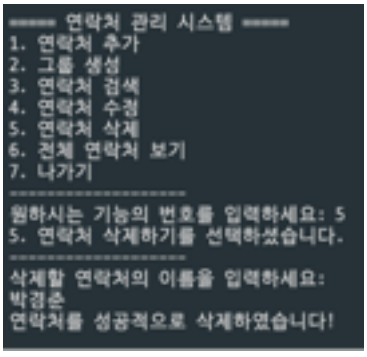
- 입력받은 이름으로 연락처를 찾고, 찾은 경우에는 해당 연락처의 정보를 사용자에게서 다시 입력받아 업데이트
    - 이름으로 연락처를 올바르게 찾을 수 있는지 테스트합니다
    - 새로 입력받아 업데이트 된 값이 제대로 출력되는지 전체 연락처 조회를 통해 테스트합니다
- 테스트 결과 스크린샷

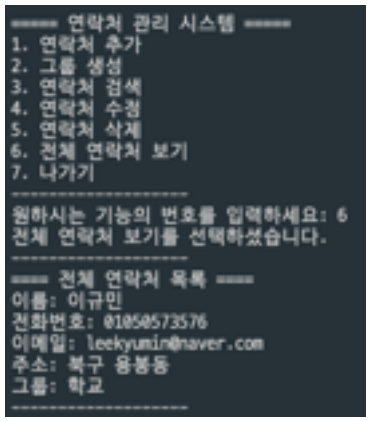




(4) Delete - 연락처 삭제 테스트

- 사용자로부터 삭제할 연락처의 이름을 입력 받음
    - 해당 연락처가 제대로 삭제되었는지 전체 연락처 조회를 통해 테스트합니다.
- 테스트 결과 스크린샷





4. 계획 대비 변경 사항

1) 변경 내역 제목

- 이전
    - 단순 CRUD 주소록 구현
- 이후
    - 검색기능을 조금 더 강화
        - 이름, 주소, 연락처, 이메일 '부분' 키워드 검색으로도 해당 연관 연락처들이 전부 검색될 수 있도록 할 것
    - 객체지향으로 분리하여 연락처가 안정적으로 저장될 수 있도록 함
    - 유효성 검사 강화
        - 모든 기능에 필요한 유효성 검사를 진행할 것
- 사유
    - 교수님 issue 반영
    - 조금 더 가독성있는 코드를 위해 객체지향적 코드를 작성해 볼 것
    - 유효성 검사의 경우, 현재 코드를 작성하고 혼자 주소록 테스트를 시도해보니, 아무런 글자, 숫자나 입력해도 그냥 작성이 되었음.
        - 웹사이트의 로그인 혹은 회원가입 폼의 경우엔 알맞은 형식을 입력해야지만 정상적으로 프로그램이 동작하였음
        - C++ 에서도 정규표현식이 javaScript 처럼 사용이 가능할 것이라고 생각됨

5. 프로젝트 일정

업무		11/26	11/27	11/29	12/02	12/03	12/06
제안서 작성		완료					
Create	필수필드입력	완료					
	추가정보입력	완료					
	유효성 검사			완료예정			
Read	전체 연락처 조회	완료					
Update	연락처 정보 수정	완료					
	유효성 검사			완료예정			
Delete	연락처 삭제 확인	완료					
	유효성 검사			완료예정			
Search	연락처 검색	진행중					
	이름으로 검색	완료					
	주소로 검색	완료					
	번호로 검색	완료					
	유효성 검사			완료예정			
유효성 검사 Test 및 리팩터링					완료예정		
객체지향 분리							완료예정